



Universidad
Zaragoza

TRABAJO FIN DE MÁSTER

Exploration of a methodology for the determination of the
mechanical properties of atheroma plaque using IVUS
imaging and Neural Networks

Exploración de una metodología para la determinación de
las propiedades mecánicas de la placa de ateroma
mediante imágenes IVUS y Redes Neuronales

Autor

Ricardo Caballero Masa

Dirigido por

Estefanía Peña Baquedano

Máster Universitario en Ingeniería Biomédica

Escuela de Ingeniería y Arquitectura

Universidad de Zaragoza

17 de Junio del 2021

Resumen

Las enfermedades cardiovasculares son la primera causa de muerte en el mundo, siendo la aterosclerosis uno de los principales motores de estas patologías. La aterosclerosis consiste en la acumulación de colesterol que traspasa el endotelio formando un núcleo lipídico muy blando que se encuentra rodeado por una capa fibrótica que lo separa del torrente sanguíneo. La rotura de esta capa provocaría la liberación del contenido del interior del núcleo y, debido a su factor trombogénico, el consiguiente riesgo de infarto de miocardio o ictus cerebral. Actualmente son la angiografía y la ecografía intravascular por ultrasonidos (IVUS) las técnicas más utilizadas para detectar esta patología, permitiendo a su vez la obtención e información relativa a las deformaciones que sufre la arteria. Este proyecto tiene como principal objetivo desarrollar una Red Neuronal Artificial (RNA) que aproveche la información de las deformaciones proporcionada por el IVUS como entradas, y prediga con un nivel de confianza suficiente las características elásticas de cada uno de los componentes de la placa de ateroma en arteria coronaria. La determinación de estas propiedades servirá, en última instancia, para ayudar en el diagnóstico de la vulnerabilidad de la placa de ateroma, así como estudiar el efecto de algunos fármacos en éstas. Para alimentar la RNA se construye una base de datos mediante el modelado 2D y el cálculo en elementos finitos de nueve geometrías con diferentes grados de estenosis. Tras un análisis estadístico de las variables más influyentes en la respuesta, se entrena y valida la RNA, cuya estructura interna ha sido previamente seleccionada mediante prueba - error. La precisión de la predicción dependerá de la estructura interna de la base de datos y de la geometría de la arteria, por lo que tener una base de datos suficientemente amplia resulta clave para el buen funcionamiento de la RNA.

Agradecimientos

Agradezco enormemente el tiempo y la sabiduría brindada durante todo este tiempo a mi tutora Estefanía Peña, al profesor Miguel Ángel Martínez, a la profesora Myriam Cilla. y a mi compañero y amigo Álvaro Tomás Latorre. Espero haber sido un buen alumno y que sigamos trabajando juntos en el futuro. También tengo palabras de gratitud para mi familia y para Rosa, mi compañera de vida. Gracias a todos por vuestro apoyo.

*Love is a better teacher
than a sense of duty.*

ALBERT EINSTEIN

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Objetivo	4
1.3. Metodología	5
2. Generación de una base de datos a partir de modelos de elementos finitos de placas vulnerables	9
2.1. Determinación de los rangos de parámetros elásticos de los componentes de la placa	9
2.2. Selección de una geometría de placa vulnerable y modelo de elementos finitos	9
2.3. Materiales	11
2.4. Cargas y condiciones de contorno	14
2.5. Mallado	15
2.6. Generación y modificación del fichero <i>.inp</i>	16
2.7. Generación y ejecución de los casos de estudio	16
3. Preprocesado y análisis estadístico	19
3.1. Análisis exploratorio de las variables	19
3.1.1. Descriptiva de ε_1 frente a E_{core} y E_{placa}	20
3.1.2. Descriptiva de ε_2 frente a E_{core} y E_{placa}	20
3.1.3. Conclusiones del análisis descriptivo de las deformaciones principales	21
3.2. Exploración de las variables más influyentes en la respuesta	22
3.2.1. Variables más influyentes en E_{core}	22
3.2.2. Variables más influyentes en E_{placa}	24
3.2.3. Análisis de regresión múltiple	26
4. Selección, generación y entrenamiento de una red neuronal	27
4.1. Selección de una estructura para la red neuronal	27
4.2. Preprocesado de datos	28

4.3. Resultados	29
4.4. Validación con geometrías reales	34
5. Conclusiones	37
5.1. Limitaciones	38
5.2. Trabajo futuro	38
Bibliografía	39
Anexos	42
A. Geometrías	43
B. Generación de la base de datos a partir de modelos de elementos finitos	49
C. Redes Neuronales	55
C.1. Concepto de Red Neuronal Artificial	56
C.2. Selección de librerías	65
C.3. Escalado de variables	66
C.4. Funciones de activación	66
D. Código completo de la Red neuronal	71

Índice de figuras

1.1. Proceso de formación y desarrollo de la placa de ateroma en arteria coronaria.	2
1.2. Imagen tomada por IVUS de una placa cuya capa fibrótica es de 36 micrómetros, su posterior segmentado manual y obtención del mapa de deformaciones.	3
1.3. Técnica basada en la elastografía iMOD para caracterizar in vivo las propiedades mecánicas de una placa aterosclerótica humana usando imágenes experimentales RF-IVUS [10].	4
2.1. Diferentes partes de una arteria con Aterosclerosis.	11
2.2. Mitad de sección transversal de una arteria coronaria idealizada con Aterosclerosis.	12
2.3. Gráfica tensión deformación de diferentes modelos constitutivos que ajustan el comportamiento de un material con comportamiento hiperelástico.	14
2.4. Condiciones de contorno de simetría, desplazamiento y presión aplicada en el lumen.	15
2.5. Mallado del modelo de arteria coronaria. Zona señalada con refinamiento de la malla por ser una zona crítica.	16
2.6. Mapa de deformaciones logarítmicas de dos casos muy diferentes: a la izquierda $E_{core} = 1kPa$ y $E_{placa} = 480kPa$, a la derecha $E_{core} = 100kPa$ y $E_{placa} = 1200kPa$	18
2.7. Modelo realista obtenido a partir de la segmentación manual de una imagen IVUS.	18
3.1. Gráfica de dispersión de la variable ε_1 frente a E_{core} a la derecha, y frente a la variable E_{placa} a la izquierda.	21
3.2. Gráfica de dispersión de la variable ε_2 frente a E_{core} a la derecha, y frente a la variable E_{placa} a la izquierda.	21
4.1. Red neuronal monocapa.	27

4.2. Gráfica de caja de las variables E_{core} y E_{placa} .	33
A.1. Tabla con el cálculo de los datos de cada sección en función del grado de estenosis.	43
A.2. Geometría y mallado correspondiente a Estenosis 40 %.	44
A.3. Geometría y mallado correspondiente a Estenosis 45 %.	44
A.4. Geometría y mallado correspondiente a Estenosis 50 %.	45
A.5. Geometría y mallado correspondiente a Estenosis 55 %.	45
A.6. Geometría y mallado correspondiente a Estenosis 60 %.	46
A.7. Geometría y mallado correspondiente a Estenosis 65 %.	46
A.8. Geometría y mallado correspondiente a Estenosis 70 %.	47
A.9. Geometría y mallado correspondiente a Estenosis 75 %.	47
A.10. Geometría y mallado correspondiente a Estenosis 80 %.	48
B.1. Distribución logarítmica de los valores que componen el estudio paramétrico.	50
B.2. Combinación de casos en el estudio paramétrico.	50
C.1. Esquema del funcionamiento de una red neuronal artificial.	57
C.2. Función de activación sigmoide	57
C.3. Funciones de activación ReLU y Softplus	58
C.4. Función no convexa con puntos de pendiente nula	59
C.5. Gráfica 3D del proceso del algoritmo descenso del gradiente.	60
C.6. Gráfico topológico 2D del proceso del algoritmo descenso del gradiente. En la figura de arriba el factor de aprendizaje es muy elevado y nunca se alcanza el mínimo error permitido. Abajo, el factor de aprendizaje es muy pequeño y el coste computacional del algoritmo es inasumible.	62
C.7. Diferentes conexiones que pueden estar relacionadas con una salida.	63
C.8. Retropropagación del error.	63
C.9. Diagrama de flujo del aprendizaje supervisado.	64
C.10. Diagrama de flujo del aprendizaje no supervisado.	64
C.11. Diagrama de flujo del aprendizaje por refuerzo.	65
C.12. Función de activación lineal.	67
C.13. Función de activación sigmoide.	67
C.14. Función de activación tangente hiperbólica.	68
C.15. Función de activación ReLU.	68
C.16. Función de activación Leaky ReLU.	68
C.17. Función de activación Elu.	69
C.18. Función de activación Selu.	70

Índice de tablas

2.1. Rangos de módulo de Young de las diferentes componentes de la arteria coronaria aterosclerótica.	10
2.2. Rangos de módulo de Young de las diferentes componentes de la arteria coronaria aterosclerótica propuestos para este trabajo. . . .	10
2.3. Tabla de factores geométricos y sus designaciones.	12
2.4. Valores de los parámetros de material utilizados en el análisis de elementos finitos para modelar la capa media y adventicia siguiendo el modelo de Gasser-Ogden-Holzapfel.	14
3.1. Resumen del modelo candidato a predecir la variable E_{core}	23
3.2. Resumen del modelo candidato a predecir la variable E_{core}	23
3.3. Resumen del modelo candidato a predecir la variable E_{core}	24
3.4. Resumen del modelo candidato a predecir la variable E_{core}	24
3.5. Resumen del modelo candidato a predecir la variable E_{placa}	24
3.6. Resumen del modelo candidato a predecir la variable E_{placa}	25
3.7. Resumen del modelo candidato a predecir la variable E_{placa}	25
3.8. Resumen del modelo candidato a predecir la variable E_{placa}	26
3.9. Resumen del modelo candidato a predecir E_{core} y E_{placa}	26
4.1. Estructura de las diferentes Redes Neuronales Artificiales analizadas.	31
4.2. Resultados de la predicción de E_{core} de las diferentes Redes Neuronales Artificiales analizadas.	32
4.3. Resultados de la predicción de E_{placa} de las diferentes Redes Neuronales Artificiales analizadas.	32
4.4. Resultados de la validación de la RNA prediciendo valores de E_{core} y E_{placa} correspondientes a nuevas geometrías no idealizadas.	35
4.5. Resultados de la validación de la RNA prediciendo valores de E_{core} y E_{placa} correspondientes a nuevas geometrías no idealizadas.	35

Capítulo 1

Introducción

1.1. Contexto

Las enfermedades cardiovasculares (ECV) son la primera causa de muerte en el mundo. Más de 17.9 millones de personas murieron de ECV en 2016, lo que representa un 31 % del total de muertes [1]. De ellas, el 85 % fueron debidas a ataques al corazón y enfermedades cerebrovasculares. En 2018, la enfermedad coronaria (EC) fue la principal causa de las muertes atribuibles a las ECV en los Estados Unidos (42,1 %), seguida de los accidentes cerebrovasculares (17,0 %) [2]. Las ECV son, por otro lado, la principal causa de la pérdida de esperanza de vida en el mundo, y la primera causa de incapacidad de la población (23 % del total de enfermedades incapacitantes). Estos datos se traducen en un coste económico de más de 169 billones de euros al año sólo en la comunidad europea [3]. La causa más común de la aparición de las ECV es la aterosclerosis y supone más de un 50 % de las muertes súbitas [4]. Ésta patología cardíaca consiste en la acumulación de células espumosas resultantes de la muerte de macrófagos que no son capaces de fagocitar todas las lipoproteínas LDL, comúnmente conocidas como colesterol malo, que traspasan el endotelio debido a un daño previo. El endotelio es la capa más interna de la arteria, y puede verse dañado por diversos factores entre los que destacan: presión arterial alta, colesterol elevado, diabetes y tabaquismo. Estos factores colaboran en que se produzca un aumento de la permeabilidad endotelial, permitiendo que las lipoproteínas LDL traspasen la pared arterial y den lugar a la acumulación de lípidos y células espumosas que forman lo que se conoce como placa de ateroma (figura 1.1).

La placa de ateroma está formada por una capa fibrótica que recubre al núcleo lipídico. Esta enfermedad suele pasar inadvertida hasta que el crecimiento de la placa de ateroma estrecha el lumen y endurece las arterias dificultando la circulación sanguínea. Uno de los mayores riesgos es que la placa de ateroma sea vulnerable,

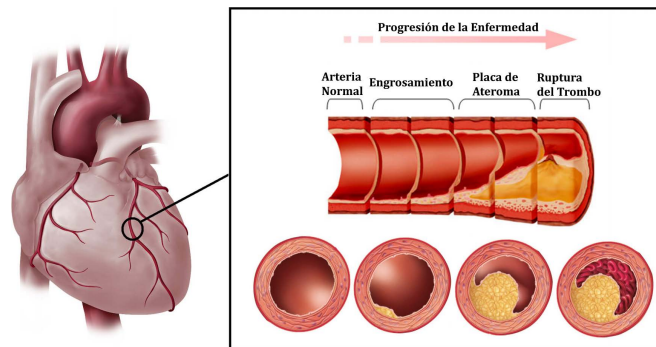


Figura 1.1: Proceso de formación y desarrollo de la placa de ateroma en arteria coronaria.

es decir, que sea susceptible de romperse liberando su contenido al torrente sanguíneo y formar un trombo que derive en un infarto de corazón o en un ictus cerebral. Para caracterizar la vulnerabilidad de una placa existen tres rasgos a considerar [5]:

- Capa fibrosa fina: se refiere a la capa fibrótica que rodea el núcleo lipídico y lo separa del torrente sanguíneo. Está constituida por colágeno, elastina y células de músculo liso. Su función es la de estabilizar y, por ello, cuanto menor sea su grosor y su contenido de colágeno, más probable es la erosión o fractura. Varios autores [6, 7, 8] defienden que este factor por sí solo es insuficiente para que se produzca la rotura.
- Amplio núcleo lipídico: el núcleo provoca un reblandecimiento de la placa y aumenta su inestabilidad. El núcleo posee factor tisular que es un gran potenciador de coagulación.
- Contenido de células inflamatorias: inducen la rotura de la placa.

A la hora de detectar enfermedades cardiovasculares, una de las técnicas más utilizadas es la angiografía, pero en ocasiones puede no detectar lesiones inestables. Es común que pacientes con angiografías normales sufran episodios coronarios [5]. Esto se debe a que este método tiende a subestimar las lesiones y, por ejemplo, placas con crecimiento positivo pasan inadvertidas. Otro inconveniente es la falta de información de la estructura de la placa, ya que mediante esta técnica sólo es visible el lumen. Por otra parte, la ecografía intravascular (IVUS) facilita información de la luz del vaso y de la pared, ya que mediante el uso de ultrasonidos obtiene imágenes de las paredes arteriales. Pero, al igual que la angiografía, se trata de un método invasivo. Sin embargo, una de las grandes ventajas que posee es la de poder visualizar las deformaciones en la pared arterial. Así pues, utilizando IVUS se han

podido desarrollar estudios [9] en los que se relacionan las zonas con altos valores de deformación tangencial con áreas propensas a la rotura y la ulceración. Por tanto, obteniendo las deformaciones radiales y tangenciales se ha observado [10] que, en placas estables, estos valores permanecen bajos. En la figura 1.2 se puede observar un ejemplo de la sección transversal de una arteria tomada por IVUS, su posterior segmentado manual [6], y la obtención del mapa de deformaciones.

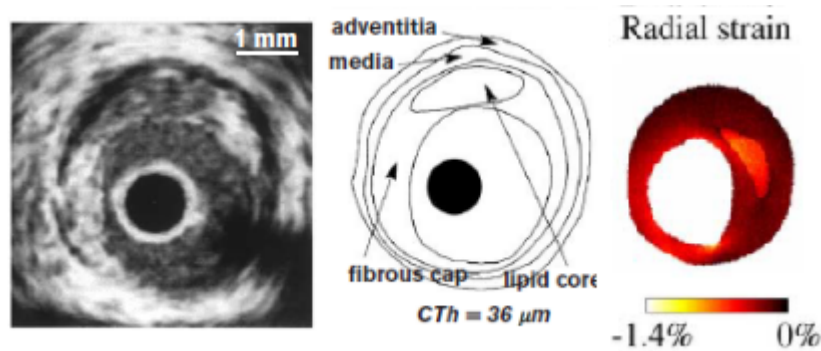


Figura 1.2: Imagen tomada por IVUS de una placa cuya capa fibrótica es de 36 micrómetros, su posterior segmentado manual y obtención del mapa de deformaciones.

En la actualidad existen diferentes técnicas para detectar placas vulnerables mediante la determinación de sus propiedades elásticas. Una de ellas es la histología virtual, basada en el análisis espectral de ultrasonidos de radiofrecuencia retrodispersados. Sin embargo, estudios realizados en *minipigs* adultos propensos a la aterosclerosis [11], mostraron que el tamaño del núcleo necrótico determinado por la histología virtual y real es diferente, cuestionando la capacidad de la histología para detectar placas propensas a la rotura. Además, esta técnica no permite una precisa cuantificación de las propiedades mecánicas de los componentes de la placa, que es esencial para el cálculo de la distribución de las tensiones. Una solución actual que tiene relación con el presente proyecto es la elastografía iMOD, que consiste en el desarrollo de varios algoritmos computacionales para intentar la reconstrucción de mapas de elasticidad dentro de la pared arterial basados en la estimación del campo de tensión obtenido con diversas técnicas derivadas del IVUS [10]. Desde 2009, se han realizado mejoras significativas como la aparición de dos poderosos biomarcadores de inestabilidad de la placa coronaria basados en las deformaciones radiales y su gradiente y que permiten la obtención, mediante técnicas de optimización, del módulo de Young bajo la hipótesis de elasticidad lineal isótropa, deformación plana e incompresibilidad [12, 13] (ver figura 1.3).

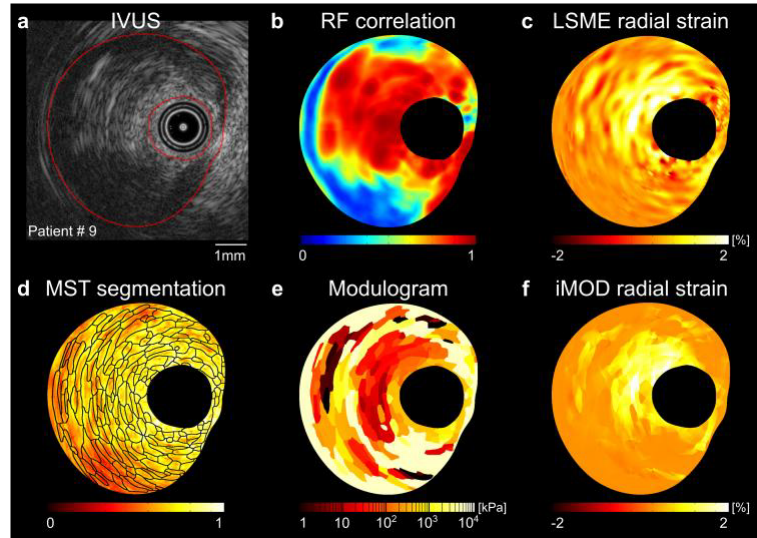


Figura 1.3: Técnica basada en la elastografía iMOD para caracterizar in vivo las propiedades mecánicas de una placa aterosclerótica humana usando imágenes experimentales RF-IVUS [10].

Todas las técnicas presentadas anteriormente plantean un problema común: el tiempo de espera desde la obtención de la imagen hasta obtener los resultados del modelo computacional son elevados debido a la necesidad de optimizar un problema para cada paciente. Este problema requiere de una alternativa que proporcione una respuesta instantánea en la detección de placas vulnerables. Para contribuir en la búsqueda de una solución, este trabajo se centra en el uso de la Inteligencia Artificial y, en concreto, las redes neuronales artificiales (RNA), para estudiar y predecir las propiedades elásticas de los componentes de la placa de ateroma con el objetivo de reducir los tiempos de cálculo y el consumo de memoria necesarios para el proceso de optimización con múltiples simulaciones de elementos finitos no lineales. Estas técnicas han sido utilizadas anteriormente por otros autores para la predicción de la vulnerabilidad de la placa de ateroma en función de la geometría de la misma, pero sin tener en cuenta las propiedades mecánicas como un factor importante en el cálculo de las tensiones [14].

1.2. Objetivo

El objetivo de este proyecto es explorar la potencialidad de las redes neuronales en la obtención de las propiedades mecánicas de la arteria coronaria en un paciente específico. En la práctica clínica, es habitual obtener medidas de deformaciones de la sección transversal de las diferentes partes de la arteria mediante imágenes

IVUS, y con ellas y la experiencia del especialista, determinar si la placa es vulnerable. Sin embargo, conseguir un conjunto de datos lo suficientemente extenso y representativo puede ser una tarea complicada y lenta. Por tanto, en este proyecto se desarrollarán modelos con diferentes geometrías mediante el software ABAQUS para generar un conjunto de datos que entrenará una RNA, simulando que han sido obtenidas mediante IVUS. Esta metodología supone una oportunidad para crear un conjunto de datos tan grande como se desee, permitiendo además estudiar cómo afectaría la variación de diferentes parámetros geométricos sin tener que esperar a conseguir una imagen real que represente dicha variación. Por último, se estudiará la influencia de las propiedades mecánicas de la placa en el diagnóstico de la aterosclerosis y de la vulnerabilidad de la placa. Todo esto, podría ayudar a obtener una respuesta instantánea en futuras intervenciones, mejorando el proceso de optimización asociado a los métodos que se emplean actualmente. Dado que este proyecto es un paso inicial en el análisis de la potencialidad de las RNA, se trabajará con algunas simplificaciones con respecto al problema real: geometría idealizada, material elástico lineal e isótropo y sin tensiones residuales. Si el resultado fuera positivo en un futuro se analizará la influencia de las hipótesis anteriormente mencionadas y se procederá a mejorar la red.

1.3. Metodología

Este proyecto se divide en dos tareas fundamentales: la generación de una base de datos a partir de modelos de elementos finitos de placas vulnerables y el desarrollo de una red neuronal para la predicción de las propiedades elásticas de los diferentes materiales que componen dichas placas. Estas tareas se enumeran a continuación.

Tarea 1. Generación de una base de datos a partir de modelos de elementos finitos de placas vulnerables. Objetivos:

- Obtención de una base de datos lo suficientemente amplia para poder entrenar una red neuronal.

Descripción del trabajo:

Subtarea 1.1. Determinación de los rangos de parámetros elásticos de los componentes de la placa.

Se ha realizado una revisión bibliográfica de propiedades de materiales y ensayos experimentales para determinar los rangos máximo y mínimo de módulos elásticos de los componentes de la placa: lípido, capa fibrótica y tejido sano. Se ha trabajado con material isótropo con el objetivo de minimizar el número de salidas de la red

neuronal. No se han incluido fenómenos como las calcificaciones.

Subtarea 1.2. Selección de una geometría de placa vulnerable y modelo de elementos finitos.

A partir de imágenes IVUS de placas de ateroma se han diseñado varias geometrías idealizadas vulnerables, con distintos grados de estenosis, que serán los modelos de partida con los que se generará la base de datos. Se ha trabajado con un modelo de elementos finitos 2D con hipótesis de deformación plana y se han despreciado las tensiones residuales con el objetivo de disminuir el tiempo de cálculo.

Subtarea 1.3. Generación de la base de datos a partir de modelos de elementos finitos.

Dada la geometría y variando los módulos elásticos dentro de los rangos determinados en la subtarea 1.1, se han realizado simulaciones de elementos finitos para obtener el campo de deformaciones en la placa simulando el proceso de adquisición mediante IVUS, es decir, con un pequeño incremento de presión y obteniendo las deformaciones relativas a la imagen inicial.

Tarea 2. Generación de una red neuronal para la predicción de las propiedades mecánicas de los componentes de la placa. Objetivos:

- Obtención de las variables de entrada de la red entre las que más influyen los módulos elásticos.
- Obtención de una red cuyos parámetros de entrada sean variables medibles con un IVUS y cuyas salidas sean los módulos elásticos de las componentes de la placa.

Subtarea 2.1. Análisis y selección de las variables de entrada para la red neuronal.

La técnica IVUS únicamente aporta información geométrica y de deformaciones, por lo que las variables candidatas para formar parte del modelo predictivo también han de serlo. Por este motivo, las variables que se han estudiado son: las deformaciones principales ($\varepsilon_1, \varepsilon_2$), la variación del espesor del núcleo (ε_{core}), la variación del espesor de la placa fibrótica (ε_{cap}) y la variación del diámetro deformado frente al indeformado (ε_{theta}). El modelo predictivo se crea y se depura mediante el análisis de la influencia de cada uno de estos parámetros en la predicción de los módulos de Young del núcleo lipídico y de la placa.

Subtarea 2.2. Selección y generación de la red neuronal.

Definidas las variables de entrada y salida y generada la base de datos, se han analizado las diferentes alternativas disponibles en literatura en materia de redes

neuronales y se ha seleccionado la más adecuada para el problema planteado. Después se llevó a cabo el proceso de entrenamiento y validación de la red seleccionada y, se estudiaron diferentes tipos de redes neuronales para encontrar aquella con menor error.

Subtarea 2.3. Validación de la herramienta mediante geometría real.

Para estudiar la influencia de la geometría en la capacidad predictiva de la herramienta desarrollada, la RNA ha de ser probada y validada en diferentes geometrías reales.

Capítulo 2

Generación de una base de datos a partir de modelos de elementos finitos de placas vulnerables

2.1. Determinación de los rangos de parámetros elásticos de los componentes de la placa

Se ha llevado a cabo una revisión del estado del arte acerca de las propiedades mecánicas de la arteria coronaria aterosclerótica. A pesar de que la información presente en la literatura a este respecto es escasa, se considera suficiente para afrontar el objetivo de este proyecto. Para mayor claridad, la información obtenida de cada estudio se ha recogido en la tabla 2.1, mostrando en cada columna los rangos considerados por los autores de dichos estudios para el núcleo lipídico, la placa y la zona sana, que abarca las capas media y adventicia de la arteria, así como la referencia de donde se han obtenido.

Con la información recogida en la tabla 2.1 y con el objetivo de cubrir un amplio rango de los módulos de Young observados en la literatura, en este trabajo se han utilizado los rangos expuestos en la tabla 2.2.

2.2. Selección de una geometría de placa vulnerable y modelo de elementos finitos

Para explicar el desarrollo del modelo de elementos finitos creado para la arteria coronaria, es necesario, en primer lugar, conocer su anatomía (ver figura 2.1). Una arteria sana está compuesta por: el lumen, es la zona por donde circula el flujo sanguíneo; capa íntima, es la capa más interna y está formada por células

Coronaria			
$E_{core}(kPa)$	$E_{placa}(kPa)$	$E_{sana}(kPa)$	Referencia
1 - 400	33 - 1000	33 - 1000	[15]
1.495	2.99 - 445.51	598	[16]
10	600	-	[12]
1	20 - 1000	10 - 800	[17]
-	50 - 1000	10 - 100	[18]
50 (1 - 202)	530 (1 - 2300)	-	[19]
5	230 - 330	80 - 170	[20]

Tabla 2.1: Rangos de módulo de Young de las diferentes componentes de la arteria coronaria aterosclerótica.

Coronaria		
$E_{core}(kPa)$	$E_{placa}(kPa)$	$E_{sana}(kPa)$
1 - 100	20 - 1200	33 - 1000

Tabla 2.2: Rangos de módulo de Young de las diferentes componentes de la arteria coronaria aterosclerótica propuestos para este trabajo.

endoteliales y sostenida por una membrana elástica interna que le aporta cierta rigidez; capa media, es la que soporta la mayoría de la presión arterial, tiene un mayor grosor y está formada por fibras musculares lisas, elastina y colágeno; y, capa adventicia, es la capa externa, está compuesta mayormente por fibroblastos y su función principal es la de proteger al vaso. Cuando aparece la aterosclerosis la anatomía de la arteria se ve ligeramente modificada, ya que aparece un núcleo lipídico muy blando y aumenta la rigidez y el tamaño de la capa íntima dando lugar a lo que se conoce como capa fibrótica.

Basándose en las imágenes que se obtienen mediante IVUS, se han diseñado y modelado nueve geometrías 2D (anexo A), idealizadas y vulnerables, que representan el rango más común de estenosis en pacientes enfermos (40 % - 80 %) [21].

El grado de estenosis se define como:

$$GradoEstenosis = \frac{Placa_{area}}{Placa_{area} + Lumen_{area}} \quad (2.1)$$

El modelo ha sido creado en ABAQUS (versión 6.14, Dassault Systems Simulia Corp., Providence, RI, USA), que es un programa de simulación que aplica el método de los elementos finitos para realizar cálculos lineales y no lineales. Además de trabajar en 2D y con el objetivo de disminuir el tiempo de cálculo, se ha considerado la hipótesis de deformación plana y se han despreciado las tensiones residuales.

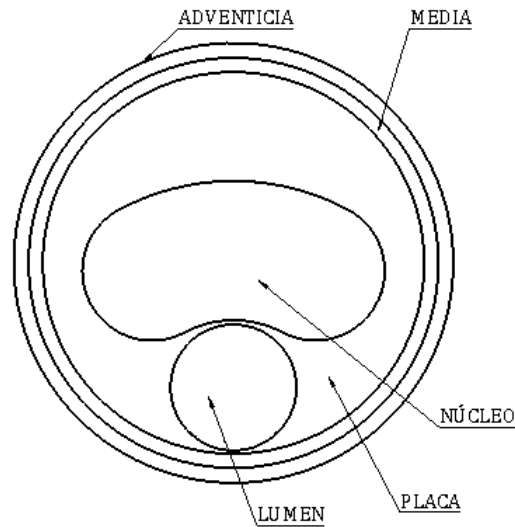


Figura 2.1: Diferentes partes de una arteria con Aterosclerosis.

Para realizar la geometría se ha partido del trabajo realizado previamente en Cilla et al. [15], donde analizaron la potencialidad de nuevos factores mecánicos para la predicción de la vulnerabilidad de la placa de ateroma. Aunque en dicho trabajo se utilizan geometrías en tres dimensiones, y teniendo en cuenta que existen estudios que han demostrado que los resultados son más precisos y fiables [13], en el presente proyecto se trabaja con modelos 2D por su mayor sencillez, puesto que el objetivo prioritario es estudiar si es posible desarrollar una herramienta que prediga la vulnerabilidad de la placa en función de sus propiedades elásticas. De este modo, se ha tomado la sección transversal de la arteria en la zona de máxima estenosis. Dado que se ha considerado una geometría idealizada, la arteria es simétrica, por lo que se puede modelar únicamente la mitad de su sección reduciendo así el coste computacional (figura 2.2). En esta sección se muestra únicamente la geometría correspondiente al grado de estenosis 40 %. El resto de geometrías pueden consultarse en el anexo A. Además, en la tabla 2.3 se puede consultar una aclaración de la nomenclatura utilizada en las figuras 2.1 y 2.2, la cual será utilizada a lo largo de todo el trabajo.

2.3. Materiales

Como se ha mencionado previamente, el objetivo es predecir el módulo de Young del núcleo lipídico y de la placa. Con este objetivo en mente, ambos materiales se han modelado como hiperelásticos, no-lineales, isótropos e incompresibles, me-

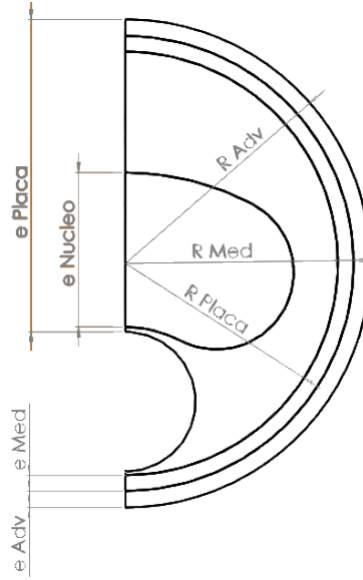


Figura 2.2: Mitad de sección transversal de una arteria coronaria idealizada con Aterosclerosis.

Radio del Lumen	R Lumen	Espesor núcleo necrótico	e Núcleo
Radio de la Placa	Placa	Espesor de la Placa	e Placa
Radio de la Media	R Med	Espesor de la Media	e Med
Radio de la Adventicia	R Adv	Espesor de la Adventicia	e Adv
Espesor Capa Fibrótica	e Cap	Ángulo Núcleo Necrótico	A Núcleo

Tabla 2.3: Tabla de factores geométricos y sus designaciones.

diante el modelo Neo-Hookeano [22]. Este modelo es el más simple y comúnmente utilizado de los modelos hiperelásticos. No obstante, previamente se ha mencionado que se asumiría la hipótesis de material elástico lineal. Estas dos ideas pueden convivir debido a que cuando se trabaja en un incremento de presión muy pequeño como el que proporciona el IVUS (5 mmHg) se puede asumir que la relación es lineal. La función de energía de deformación general se puede escribir de la siguiente manera:

$$W = C_1(I_1 - 3) + D_1(J - 1)^2 \quad (2.2)$$

Donde:

- W es la densidad de energía de deformación
- C_1 es un parámetro del material que está relacionado con el módulo de

cizalladura. Para deformaciones pequeñas como las que se van a experimentar y para material incompresible puede expresarse en función del módulo de Young como $C_1 = E/6$ (ver figura 2.3)

- I_1 es el primer invariante (traza) del tensor de deformación de Cauchy - Green por la izquierda: $I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2$ donde λ_i son las deformaciones principales
- J es el Jacobiano de la matriz de deformaciones, representa la deformación volumétrica y se calcula como el determinante del gradiente de deformación ($\det \nabla F$)
- El gradiente de deformación se expresa como $\nabla F = \frac{\partial x}{\partial X}$ y representa la derivada parcial de las coordenadas en la configuración deformada respecto a la indeformada
- D_1 es un parámetro del material que representa su incompresibilidad

De este modo, asumiendo la consideración de que el material es incompresible, no habrá una deformación volumétrica ($J = 1$) y el factor de incompresibilidad D_1 será cero. Así, la ecuación 2.2 se simplifica en la ecuación 2.3:

$$W = C_1(I_1 - 3) \quad (2.3)$$

La figura 2.3 representa la gráfica tensión deformación de varios modelos constitutivos que ajustan el comportamiento de un material hiperelástico [21], y puede verse que el modelo Neo-Hookeano representa el comportamiento lineal.

Por otro lado, se ha considerado que las capas media y adventicia no están enfermas, y dado que no forman parte de los parámetros de interés a predecir por la RNA, su comportamiento se ha modelado con un modelo hiperelástico algo más complejo. El modelo seleccionado ha sido el de Gasser [23], empleado para materiales anisótropos, el cual considera dos familias de fibras, pero con un mismo comportamiento, donde k_1 corresponde a la rigidez general del material y k_2 es la rigidez de las fibras de colágeno. Igual que en el modelo Neo-Hookeano, el parámetro D se refiere a la incompresibilidad ($D=0$ para considerarlo incompresible). El parámetro μ es la rigidez de la elastina, y k aporta información sobre la dispersión de las fibras (se considera $k=0.33$ cuando el material es isótropo porque significa que las fibras se distribuyen sin seguir una dirección preferente, por lo que el comportamiento del material será el mismo en cualquier dirección). El ángulo de las fibras está definido respecto de la dirección circunferencial.

$$\Psi = \frac{1}{D}(J-1)^2 + \mu(I_1-3) + \frac{k_1}{2k_2} \sum_{i=4,6} \exp(k_2(k(I_1-3) + (I_1-3k)(I_i-1))) - 1 \quad (2.4)$$

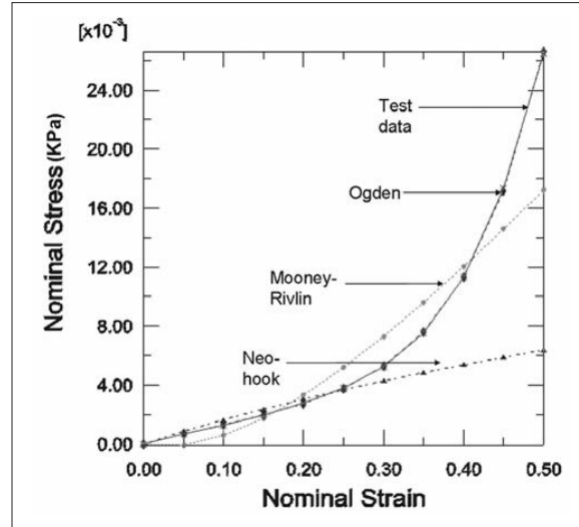


Figura 2.3: Gráfica tensión deformación de diferentes modelos constitutivos que ajustan el comportamiento de un material con comportamiento hiperelástico.

Los valores numéricos que toma cada una de las variables que conforman este modelo son los mismos que se utilizaron en Cilla et al. [15].

	μ (kPa)	k_1 (kPa)	k_2 (-)	k (-)
Adventicia	8.44	547.67	568.01	0.26
Media	1.4	206.16	58.55	0.29

Tabla 2.4: Valores de los parámetros de material utilizados en el análisis de elementos finitos para modelar la capa media y adventicia siguiendo el modelo de Gasser-Ogden-Holzapfel.

2.4. Cargas y condiciones de contorno

Para el correcto análisis del modelo, es necesario establecer las cargas y las condiciones de contorno del problema. Se ha trabajado con una situación desfavorable pero dentro de los límites de lo común, por tanto la presión arterial corresponde a la presión alta (140 mmHg). Dado que se trabaja en kPa, se introduce como carga 18,87 kPa. Las condiciones de contorno son las siguientes: simetría en el eje X y restricción del movimiento en el eje Y en un punto de la periferia (ver figura 2.4). De este modo se está contemplando que se trabaja únicamente sobre la mitad simétrica del modelo y se evita el desplazamiento y el giro del mismo.

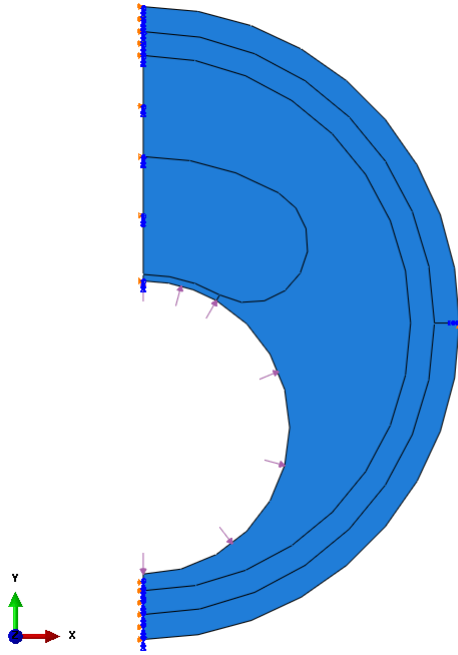


Figura 2.4: Condiciones de contorno de simetría, desplazamiento y presión aplicada en el lumen.

2.5. Mallado

Para alcanzar un equilibrio entre buena precisión y coste computacional asumible, se ha decidido implementar una malla con una separación entre nodos de 0.05 mm en toda la placa, excepto en la zona de la capa fibrótica, que se ha establecido una separación entre nodos de 0.01 mm (ver figura 2.5). Esto es debido a que la capa fibrótica es la zona de mayor interés, ya que se sabe que por ser la zona más estrecha entre el núcleo lipídico y el lumen ($65\mu m$), es una zona de concentración de tensiones y por tanto es mucho más probable que se produzca la rotura de la placa. Además, dado que el material se ha considerado incompresible, es necesario utilizar la formulación híbrida en ABAQUS, que trata de forma desacoplada la parte volumétrica y la desviadora. En total, se ha generado una malla compuesta por 5192 nodos y 4989 elementos del tipo CPE4H.

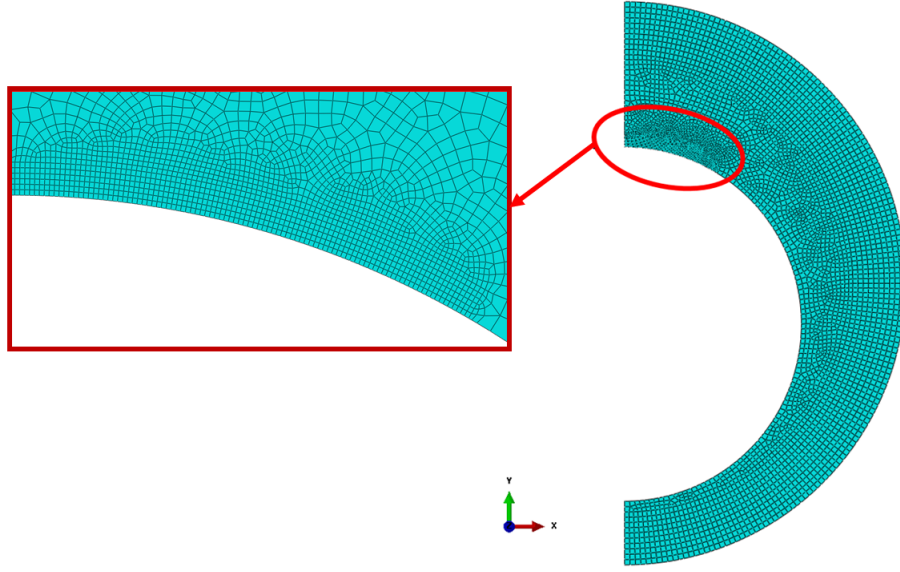


Figura 2.5: Mallado del modelo de arteria coronaria. Zona señalada con refinamiento de la malla por ser una zona crítica.

2.6. Generación y modificación del fichero *.inp*

Una vez se ha creado el modelo, se genera el fichero *.inp*, que es un fichero de texto que contiene toda la información del modelo creado en ABAQUS. Al disponer de esta información en formato de texto, se puede modificar fácilmente sin tener que entrar a la interfaz gráfica. De este modo, es más sencillo parametrizar los valores correspondientes a las propiedades del material. El objetivo detrás de la parametrización de este fichero es poder crear un único *Script* en Python que permita crear y ejecutar de forma rápida y sencilla todos los casos de interés para generar la base de datos. Dicha base de datos estará compuesta por los resultados de las simulaciones de cada configuración. Las variables a parametrizar son aquellas que corresponden al material del núcleo y de la placa: el núcleo se parametriza como $C_{core} = \frac{E_{core}}{6}$ y la placa como $C_{placa} = \frac{E_{placa}}{6}$. De este modo, creando un estudio paramétrico en el que se varíen los parámetros E_{core} y E_{placa} , se podrán generar tantos casos como se desee para crear la base de datos.

2.7. Generación y ejecución de los casos de estudio

Para crear la base de datos, una vez se ha parametrizado el fichero *.inp*, es necesario crear un *Script* que permita a las variables parametrizadas tomar diferentes valores numéricos que creen tantas configuraciones como casos de estudio interesen.

ABAQUS recoge esta metodología en su módulo *Estudios Paramétricos*, mediante el cual se puede generar, ejecutar y presentar los resultados de múltiples análisis que difieren únicamente en las variables parametrizadas. De este modo, se evita tener que especificar manualmente el valor numérico de la variable en cada caso de interés. Las instrucciones específicas que componen el *Script* para realizar el estudio paramétrico, se pueden consultar en el anexo B.

Como se verá en la sección 3.1, este trabajo ha tenido dos fases en cuanto a la generación de los casos de estudio se refiere, variando el módulo de Young de cada componente de la placa atendiendo a los rangos obtenidos en el apartado 2.2. En una primera fase inicial, se consideró necesario realizar 400 casos distribuidos normalmente, mientras que en otra fase posterior, tras analizar los resultados obtenidos, se observó la necesidad de ampliar la base de datos a 30000 casos con una distribución logarítmica. En la figura 2.6 se pueden observar las deformaciones logarítmicas principales de la deformada para dos casos de estudio muy diferentes en cuanto a las propiedades elásticas de sus componentes. En estas dos muestras se puede observar que, cuanta mayor rigidez tienen los materiales, menor deformación experimentan:

- Caso 1: $E_{core} = 1kPa$ y $E_{placa} = 480kPa$
- Caso 2: $E_{core} = 100kPa$ y $E_{placa} = 1200kPa$

Para validar los resultados de la red, se ha contado con una geometría real obtenida mediante IVUS, segmentada manualmente por un experto [12] y modelada posteriormente en ABAQUS (figura 2.7).

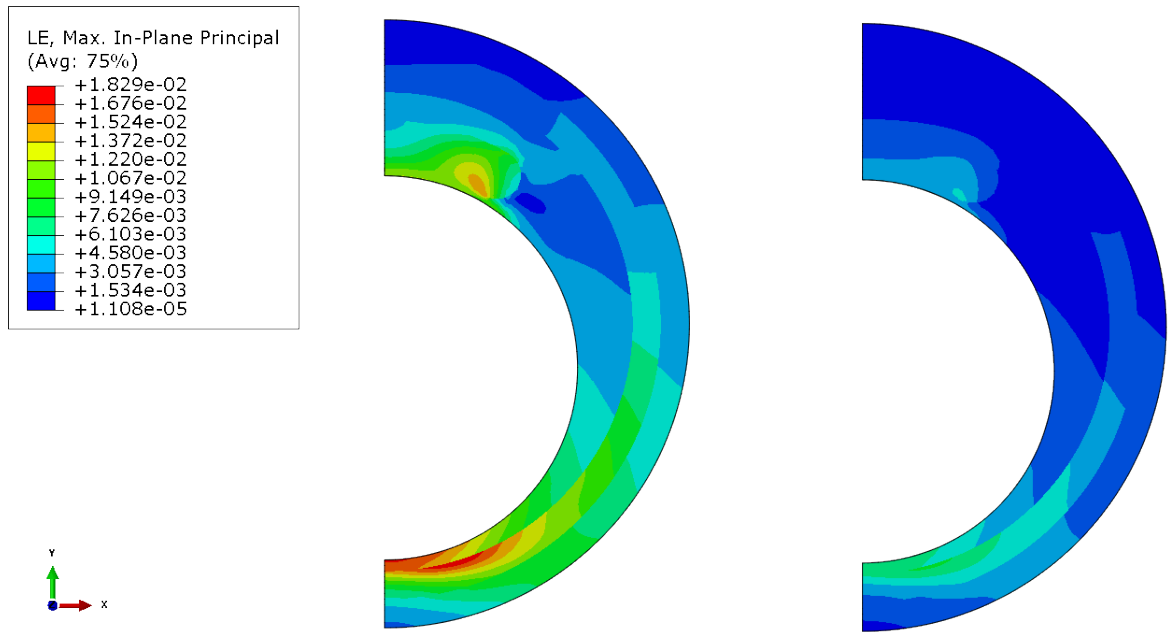


Figura 2.6: Mapa de deformaciones logarítmicas de dos casos muy diferentes: a la izquierda $E_{core} = 1kPa$ y $E_{placa} = 480kPa$, a la derecha $E_{core} = 100kPa$ y $E_{placa} = 1200kPa$

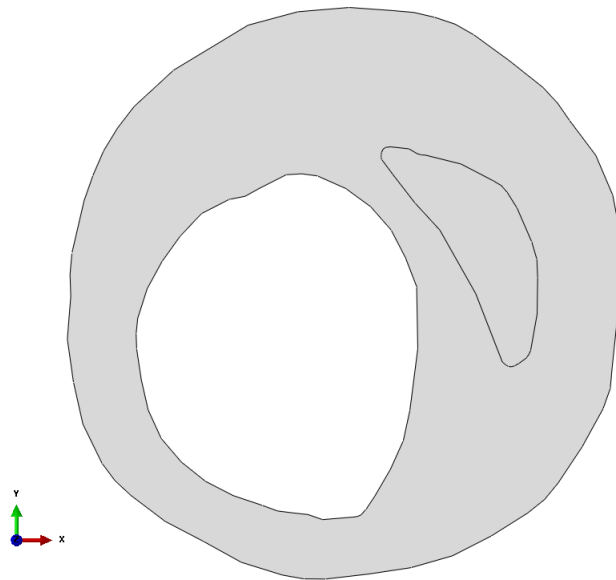


Figura 2.7: Modelo realista obtenido a partir de la segmentación manual de una imagen IVUS.

Capítulo 3

Preprocesado y análisis estadístico

La Inteligencia Artificial basa su capacidad de predicción en la estadística. Por tanto, es necesario realizar un análisis que evidencie cuáles son las variables más influyentes sobre la respuesta, para así encontrar el mejor modelo predictivo. Este análisis se ha dividido en dos partes: análisis descriptivo de las variables y obtención del modelo predictivo. Las variables candidatas a explicar la respuesta estarán siempre relacionadas con las deformaciones o con la geometría, ya que es la información que se puede extraer del IVUS:

- ε_1 : deformación principal máxima.
- ε_2 : deformación principal mínima.
- ε_{core} : variación del espesor del núcleo con respecto a su espesor inicial.
- ε_{cap} : variación del espesor de la capa fibrótica fina que separa el núcleo del torrente sanguíneo con respecto a su espesor inicial.
- ε_{theta} : variación del diámetro del lumen con respecto a su diámetro inicial.
- $Estenosis(\%)$: relación entre el área de la placa y el área del conjunto placa más lumen. Inicialmente esta variable no fue considerada, pero tras analizar los primeros resultados, fue añadida.

3.1. Análisis exploratorio de las variables

En el análisis exploratorio de datos, una primera aproximación consiste en reducir toda la información muestral a un reducido grupo de valores numéricos, los parámetros estadísticos, cuyo conocimiento puede dar pie a la detección de algún patrón de comportamiento que aporte información sobre la distribución de probabilidad bajo la que se ha obtenido la muestra. Cabe destacar que este análisis

se ha realizado dos veces a lo largo del proyecto. La primera corresponde a una fase prematura en la que se contaba apenas con 400 datos, y en ella se analizaron las variables de deformaciones principales ε_1 y ε_2 , así como las variables ε_{core} , ε_{cap} y ε_{theta} . A pesar de que los modelos de regresión multivariada eran buenos (R-cuadrado ajustado del 92 %), la red neuronal no predecía correctamente los valores de E_{core} , sobre todo en los valores cercanos al cero, incurriendo en errores de más del 500 %. Dado que E_{placa} se predecía con errores por debajo del 1 %, se llegó a la conclusión de que el experimento estaba mal diseñado, pues el amplio rango de valores que tomaba E_{core} (1 kPa - 400 kPa) suponía en una gran variabilidad en comparación con el rango de valores que tomaba E_{placa} (390 kPa - 1200 kPa). De este modo, se tomaron dos decisiones. Por un lado, revisar la literatura para ver cuantos artículos cubrían el rango 1 kPa - 400 kPa, y se descubrió que algunos artículos [15], [17] habían llegado a la conclusión de que a partir de 100 kPa el núcleo se consideraba tan rígido que se confundía con la placa. Por tanto, se redujo el rango a 1 kPa - 100 kPa. Por otro lado, para solventar el problema de las malas predicciones en valores cercanos al cero, se añadieron más casos de estudio en el rango 1 kPa - 20 y muchos menos en el resto. Por este motivo, y dado que en el análisis previo se observó que las variables ε_1 y ε_2 no eran lo suficientemente significativas, se decidió repetir el análisis estadístico con los 30000 datos resultantes únicamente para las variables ε_{core} , ε_{cap} y ε_{theta} .

3.1.1. Descriptiva de ε_1 frente a E_{core} y E_{placa}

Como se observa en la figura 3.1, el valor de ε_1 disminuye conforme aumenta E_{core} . Esta relación era de esperar dado que cuanto más rígido sea el núcleo, menor capacidad tendrá para deformarse. La misma tendencia se sigue al representar ε_1 frente a E_{placa} : cuanto más rígido sea el material que rodea al núcleo menor será su libertad para deformarse. Además, ambas figuras muestran una relación no lineal, siendo esta información de utilidad al generar el modelo: será necesario un modelo no lineal.

3.1.2. Descriptiva de ε_2 frente a E_{core} y E_{placa}

Al representar ε_2 frente a E_{core} y E_{placa} se observa que, siguiendo el mismo criterio que ε_1 , la variable ε_2 disminuye conforme aumentan los valores de E_{core} y E_{placa} , y además las gráficas de dispersión muestran un comportamiento no lineal.

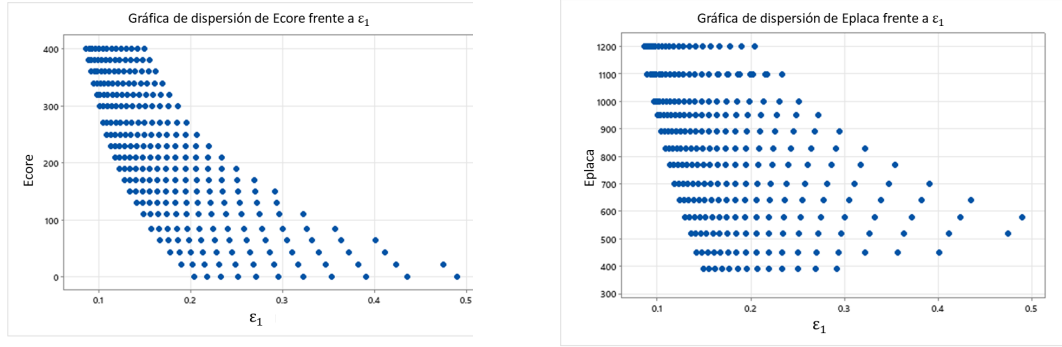


Figura 3.1: Gráfica de dispersión de la variable ε_1 frente a E_{core} a la derecha, y frente a la variable E_{placa} a la izquierda.

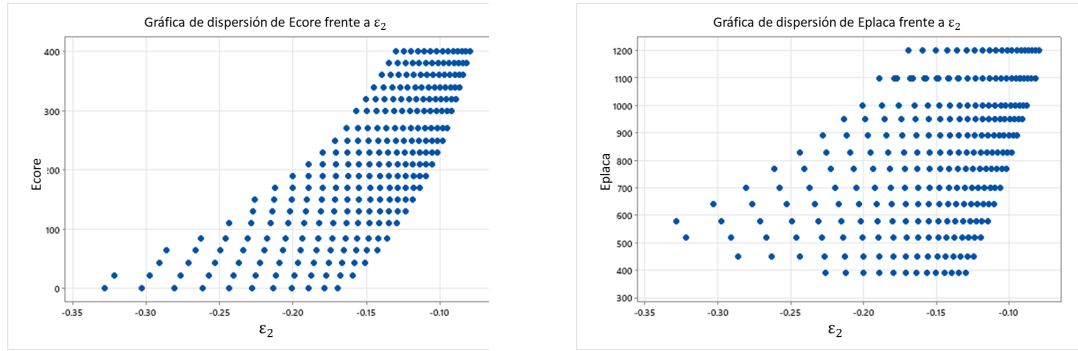


Figura 3.2: Gráfica de dispersión de la variable ε_2 frente a E_{core} a la derecha, y frente a la variable E_{placa} a la izquierda.

3.1.3. Conclusiones del análisis descriptivo de las deformaciones principales

Las conclusiones de no linealidad y de menor deformación cuanto más rígidos son los materiales se pueden extrapolar al resto de variables. Representarlas gráficamente puede llevar a confusión por la gran cantidad de datos que se maneja, y se estaría incurriendo en la reiteración de las mismas conclusiones.

3.2. Exploración de las variables más influyentes en la respuesta

Una vez se conoce que las variables con las que se trabaja presentan comportamientos no lineales, se va a buscar un modelo que incluya términos no lineales para ser capaces de explicar la respuesta con mayor precisión. Aunque el objetivo final es encontrar el modelo que mejor predice dos salidas simultáneas, E_{core} y E_{placa} , se han realizado varios análisis de regresión lineal múltiple con una única variable de salida previamente, para encontrar cuáles son aquellas variables que tienen mayor influencia por separado en cada una de las salidas. Para conseguir el mejor modelo, se utiliza en todo el análisis el método paso a paso [24], con un valor de α de entrada y de salida de 0.05. La forma de operar de este método es la siguiente: se crea un modelo que incluye una única variable, después se hace una iteración en la que sólo se incluirá la siguiente variable si mejora la respuesta del modelo, y así con todas las variables candidatas. Que el α sea 0.05 implica que la variable solo será incluida cuando su p-valor sea menor que 0.05. Por tanto existen dos opciones: si el p-valor es menor que 0.05 se considera que la variable es influyente en la respuesta con un nivel de confianza del 99.95 % y se introduce en el modelo; en caso contrario, se excluye del modelo. Por otro lado, se ha realizado un análisis de regresión múltiple multivariada para estudiar cuál podría ser un buen modelo cuando se trata de predecir dos variables respuesta de forma simultánea. La metodología seguida para este propósito ha sido la misma que para el análisis de regresión múltiple para una única variable respuesta. De este modo, y como puede observarse en detalle en los siguientes apartados, se ha averiguado que las variables con mayor influencia son: $\varepsilon_{core-cuad}$, $\varepsilon_{cap-cuad}$, $\varepsilon_{theta-cuad}$ y $E_{stenosis}(\%)$.

3.2.1. Variables más influyentes en E_{core}

En un primer análisis, se incluyeron todas las variables candidatas y se encontró que el modelo era capaz de explicar la respuesta correctamente un 99.64 % de las veces (tabla 3.1), donde la *Constante* que aparece en dichas tablas es la ordenada en el origen de la recta de regresión. Sin embargo, el parámetro FIV, el cuál representa la multicolinealidad del modelo, es muy elevado, es decir, hay variables que están explicando la misma parte de la respuesta, por lo que son prescindibles. Además, la variable ε_2 está siendo poco influyente en la respuesta ($p\text{-valor} > 0.05$), por lo que se excluyó del modelo.

Para depurar y mejorar el modelo, fue necesario reducir el número de variables. Dado que se observó que el problema era no lineal, se excluyeron aquellas variables que no estaban elevadas al cuadrado (excepto la estenosis que es una variable clave de interés en esta investigación). Al eliminarlas el valor que indica la multi-

	Coef	Valor p	FIV
<i>Constante</i>	-1476095	0,000	
ε_1	-31301	0,000	68225,45
ε_{core}	-16730	0,000	105604,65
ε_{cap}	-163720	0,000	459652,78,45
ε_{theta}	1270997	0,000	$9,3416 \cdot 10^8$
<i>Estenosis</i> (%)	26244	0,000	$7,0147 \cdot 10^8$
$\varepsilon_{core-cuad}$	12059	0,000	125086,38
$\varepsilon_{cap-cuad}$	70125	0,000	159945,58
$\varepsilon_{theta-cuad}$	-440801	0,000	16945726,55
Resumen del modelo			
R-cuadrado	99,65 %		
R-cuadrado (ajustado)	99,64 %		

Tabla 3.1: Resumen del modelo candidato a predecir la variable E_{core} .

colinealidad del problema (FIV) disminuyó drásticamente, y el valor que indica la precisión del modelo (R-cuad ajustado) se mantuvo en 92.80 %. Sin embargo, el método paso a paso eliminó la variable $\varepsilon_{theta-cuad}$, considerándola poco influyente en la predicción de E_{core} (tabla 3.2).

	Coef	Valor p	FIV
<i>Constante</i>	2151	0,000	
<i>Estenosis</i> (%)	-47,03	0,000	2,52
$\varepsilon_{core-cuad}$	-2899	0,000	73,06
$\varepsilon_{cap-cuad}$	4049	0,000	83,39
Resumen del modelo			
R-cuadrado	92,88 %		
R-cuadrado (ajustado)	92,80 %		

Tabla 3.2: Resumen del modelo candidato a predecir la variable E_{core} .

La métrica FIV, aunque disminuyó drásticamente, seguía siendo elevada. De este modo, eliminando la variable $\varepsilon_{cap-cuad}$ se observó que la precisión del modelo disminuía al 83,6 % (tabla 3.3), mientras que si se eliminaba la variable $\varepsilon_{core-cuad}$, el modelo explicaba el 90 % (tabla 3.4) y se conseguía disminuir el FIV a 1,79, que es un valor aceptable.

	Coef	Valor p	FIV
<i>Constante</i>	-161,3	0,022	
<i>Estenosis</i> (%)	-30,98	0,000	1,57
$\varepsilon_{core-cuad}$	2321,6	0,000	1,57
Resumen del modelo			
R-cuadrado	83,72 %		
R-cuadrado (ajustado)	83,60 %		

Tabla 3.3: Resumen del modelo candidato a predecir la variable E_{core} .

	Coef	Valor p	FIV
<i>Constante</i>	893,4	0,000	
<i>Estenosis</i> (%)	-39,32	0,000	1,79
$\varepsilon_{cap-cuad}$	1849,1	0,000	1,79
Resumen del modelo			
R-cuadrado	90,12 %		
R-cuadrado (ajustado)	90,04 %		

Tabla 3.4: Resumen del modelo candidato a predecir la variable E_{core} .

3.2.2. Variables más influyentes en E_{placa}

Siguiendo la misma metodología que en el anterior apartado, se introdujeron, en primer lugar, todas las variables candidatas que conforman el conjunto de datos, y se creó un modelo de regresión paso a paso que incluyera o excluyera variables del modelo con un nivel de significancia $\alpha = 0,05$. El propio método eliminó ε_2 , ε_{1-cuad} , ε_{2-cuad} , ε_{core} , ε_{cap} y $\varepsilon_{cap-cuad}$, pero debido a que el factor de multicolinealidad era muy elevado fue necesario depurar el modelo (ver tabla 3.5).

	Coef	Valor p	FIV
<i>Constante</i>	3740248	0,000	
ε_1	-1437	0,000	113,40
ε_{theta}	-3015655	0,000	$8,0239 \cdot 10^8$
<i>Estenosis</i> (%)	-62235	0,000	$6,02486 \cdot 10^8$
$\varepsilon_{core-cuad}$	-2875	0,000	102,15
$\varepsilon_{theta-cuad}$	1047476	0,000	$14,5612 \cdot 10^6$
Resumen del modelo			
R-cuadrado	99,56 %		
R-cuadrado (ajustado)	99,56 %		

Tabla 3.5: Resumen del modelo candidato a predecir la variable E_{placa} .

Al crear un nuevo modelo, introduciendo únicamente como variables candidatas aquellas que están al cuadrado, el grado de estenosis por ser variable de especial interés, y ε_1 por haber sido influyente en el modelo anterior con un FIV relativamente bajo, se obtuvo que: el factor de multicolinealidad disminuyó notablemente; el modelo eliminó la variable ε_1 ; y la precisión del modelo seguía siendo considerablemente elevada.

	Coef	Valor p	FIV
<i>Constante</i>	-3908	0,000	
<i>Estenosis</i> (%)	119,11	0,000	2,52
$\varepsilon_{core-cuad}$	-2839	0,000	73,06
$\varepsilon_{cap-cuad}$	1315	0,000	83,39
Resumen del modelo			
R-cuadrado	97,91 %		
R-cuadrado (ajustado)	97,88 %		

Tabla 3.6: Resumen del modelo candidato a predecir la variable E_{placa} .

A continuación, se muestran dos pruebas adicionales para comprobar si era posible disminuir el FIV:

- Modelo que incluye las variables *Estenosis*(%) y $\varepsilon_{core-cuad}$:

	Coef	Valor p	FIV
<i>Constante</i>	-4658,4	0,000	
<i>Estenosis</i> (%)	124,32	0,000	1,57
$\varepsilon_{core-cuad}$	-1144,1	0,000	1,57
Resumen del modelo			
R-cuadrado	97,67 %		
R-cuadrado (ajustado)	97,65 %		

Tabla 3.7: Resumen del modelo candidato a predecir la variable E_{placa} .

- Modelo que incluye las variables *Estenosis*(%) y $\varepsilon_{cap-cuad}$:

De este modo, se comprobó que ambos eran influyentes. Sin embargo, si se encontraban ambos a la vez el modelo no mejoraba, por lo que podrían estar explicando la misma parte de la respuesta.

	Coef	Valor p	FIV
<i>Constante</i>	-5139,1	0,000	
<i>Estenosis</i> (%)	126,67	0,000	1,79
$\varepsilon_{cap-cuad}$	-840	0,000	1,79
Resumen del modelo			
R-cuadrado	97,26 %		
R-cuadrado (ajustado)	97,24 %		

Tabla 3.8: Resumen del modelo candidato a predecir la variable E_{placa} .

3.2.3. Análisis de regresión múltiple

Por último, para estudiar cual sería el mejor modelo para predecir ambas respuestas de forma simultánea, se realizó un análisis de regresión múltiple. En él, se introdujeron como variables candidatas a formar parte del modelo: *Estenosis*(%), $\varepsilon_{core-cuad}$, $\varepsilon_{cap-cuad}$ y $\varepsilon_{theta-cuad}$. Con estas variables, la precisión del modelo alcanzó un 92.80 %, y la tabla de coeficientes indicaba que la variable $\varepsilon_{theta-cuad}$ no era significativa. No obstante, cabe destacar que, este análisis era orientativo, y que la red neuronal podría predecir mejor incluyendo alguna de las variables eliminadas. Esto se analiza en mayor profundidad en el siguiente capítulo.

	Coef	Valor p
<i>Constante</i>	2540	0,000
<i>Estenosis</i> (%)	-53,62	0,000
$\varepsilon_{core-cuad}$	-2978	0,000
$\varepsilon_{cap-cuad}$	4109	0,000
Resumen del modelo		
R-cuadrado	92,91 %	
R-cuadrado (ajustado)	92,80 %	

Tabla 3.9: Resumen del modelo candidato a predecir E_{core} y E_{placa} .

Capítulo 4

Selección, generación y entrenamiento de una red neuronal

4.1. Selección de una estructura para la red neuronal

La estructura más simple de una red neuronal es aquella formada únicamente por una capa de entrada y una capa de salida. El hecho de añadir capas ocultas mejorará la capacidad de aprendizaje de la red. Sin embargo, añadir demasiadas capas ocultas puede resultar en un sobreentrenamiento de la red, también llamado *Overfitting*.

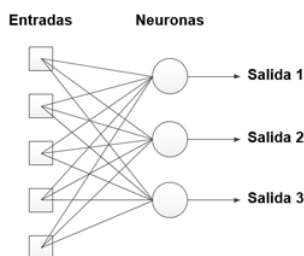


Figura 4.1: Red neuronal monocapa.

Para crear la red se ha utilizado la librería de tensorflow `keras.sequential()`. Esta librería precisa del número de neuronas por capa, de una semilla de datos para empezar a realizar las iteraciones y de una función de activación en cada capa. La función `add()` se utiliza para añadir capas a la red neuronal. Se han utilizado tres tipos de capas: capa densa (`dense()`), es decir, cada neurona de la capa actual se conecta con todas las neuronas de la capa siguiente; `dropout()`, que anula algunas neuronas de forma aleatoria en cada iteración para no aprender de todas ellas

al mismo tiempo, protegiendo así a la red de sufrir (*overfitting*); y una capa de normalización por lotes (`BatchNormalization()`), que normaliza las salidas de la capa anterior para que evitar el efecto predominante cuando se utilizan parámetros que no están en la misma escala. El número de neuronas de salida ha de ser igual al número de variables respuesta que se quiera predecir, en este caso, serán dos: E_{core} y E_{placa} . Además, es necesario determinar algunos parámetros adicionales:

- Optimizador: se encarga de minimizar el error cometido por la red
- Número de épocas: es el número de iteraciones que realiza el algoritmo para ajustar los pesos de cada neurona
- Tamaño de lote: número de instancias de entrenamiento que el modelo puede ver durante una iteración de entrenamiento. En lugar de ajustar los pesos de cada neurona de manera individual, se ajustan en lotes para ayudar a no sobre entrenar la red y alcanzar un menor error.

Una vez creada una estructura provisional, se procede al entrenamiento de la red. El entrenamiento finalizará tras realizar el número de iteraciones determinadas fijadas llamadas *número de épocas*. La red se dará por entrenada cuando el error obtenido mediante la métrica del error cuadrático medio sea lo suficientemente bajo. Después se analiza que tal predice aquellos datos que fueron reservados para probar la herramienta, y no han sido utilizados para el entrenamiento. Una vez que el error de predicción es lo suficientemente bajo en estos datos de prueba ($error < 20\%$) se considera que se ha alcanzado una buena red (ver código completo en anexo D).

4.2. Preprocesado de datos

El conjunto de datos obtenido del estudio paramétrico es dividido en dos subconjuntos, uno de entrenamiento (80 %) y otro de *testing* (20 %). Los datos de entrada son diferentes, y por tanto pueden tener órdenes de magnitud distintos. Este es el caso por ejemplo entre los parámetros ε_{core} (toma valores desde 10^{-5} hasta 10^{-2}) y grado de estenosis (toma valores desde 40 % hasta el 80 %). Esto puede suponer un problema a la hora de comparar elementos, ya que el peso del grado de estenosis podría anular el efecto que tiene la deformación del núcleo lipídico (anexo C.3). Para solventar este inconveniente, el escalado de variables es una práctica fundamental en el análisis de datos. Se ha utilizado un escalado Máximo – Mínimo [25] (ecuación 4.1), cuya función es transformar el valor más pequeño a cero y el más alto a uno. El resto quedan entre estos valores de tal forma que se mantengan las proporciones entre ellos.

$$y = \frac{x - min}{max - min} \quad (4.1)$$

4.3. Resultados

Para encontrar la estructura de la red que mejor prediga se ha seguido una metodología de prueba-error. Sin embargo, debido a la infinidad de combinaciones y pruebas que podrían realizarse, el entendimiento de lo explicado hasta ahora es crucial para poder encontrar una estructura adecuada. En este apartado se van a explicar las diferentes pruebas realizadas en la búsqueda de la mejor estructura de la RNA, y se van a ordenar de forma cronológica en la tabla 4.1. Además, los resultados que proporcionaron dichas redes neuronales se muestran en las tablas 4.2 y 4.3 .

En un primer intento se creó una RNA con pocas capas, con el objetivo de intentar mantener un coste computacional bajo, y se utilizó el conjunto de datos reducido mencionado en la sección 3.1, que cuenta con 400 datos correspondientes a la base de datos generada con la geometría de grado de estenosis 40 %. De este modo, se seleccionaron cuatro capas con 50, 25, 10 y 2 neuronas en cada capa respectivamente. En este momento del estudio, la variable *Estenosis*(%) aún no se había introducido, dado que únicamente se disponía de una geometría. El resultado fue un error del 100 % en E_{core} . Esto era un indicativo de que algo no estaba funcionando correctamente, y se cree que el motivo podría deberse a la función de activación seleccionada, pues la función ReLU no es derivable en cero y en este trabajo se manipulan valores muy cercanos al cero. El resultado del error en la predicción de E_{placa} era mucho mejor, obteniendo un error medio del 3.5 %.

Teniendo en cuenta lo anterior, el segundo intento se centró en encontrar la función de activación óptima para la tipología de los datos que se estaban manipulando. De todas las candidatas (ver anexo C.4) se seleccionaron aquellas parecidas a la ReLU pero que pudieran tomar valores distintos de cero cuando una entrada fuera negativa. Aunque se observaron algunas mejoras, ninguna daba resultados suficientemente buenos, cometiendo errores de más del 1000 %.

- Elu: error medio de E_{core} del 1561 % y de E_{placa} del 4.85 %.
- Selu: error medio de E_{core} del 1672 % y de E_{placa} del 6.92 %.
- Softplus: error medio de E_{core} del 1578 % y de E_{placa} del 4.2 %.

En la tercera prueba se decidió aumentar el número de capas y de neuronas. Tras varias pruebas se halló una combinación que ofrecía resultados con errores relativamente bajos en todos los casos excepto cuando se trataba de predecir valores de E_{core} cercanos a 1 kPa. Analizando los datos predichos, se observa que el error promedio de E_{core} es 55.74 %, y en E_{placa} es 2.89 %. Esto supone una gran mejoría

respecto a las pruebas previas. No obstante, el error en E_{core} aun no es aceptable, pues la mayoría de las predicciones de valores cercanos a 1 kPa las hace con un valor concreto que era negativo (-1.75 kPa). Por este motivo, se cree que la función de activación Softplus podría ser la solución, ya que nunca toma valores de predicción negativos, aunque sí se activa con valores de entrada negativos, a diferencia de la ReLU. Además, se realizó una búsqueda de los mejores parámetros para la red, con el comando **GridSearch**. Este comando permite al usuario probar, de una sola vez, distintos parámetros de la red como diferentes optimizadores para , número de épocas y tamaño de lote. Tras realizar esta prueba se observó que lo que ofrecía un mejor comportamiento era: optimizador Adam [26], [27], número de épocas igual a 300, y tamaño de lote igual a 8.

La cuarta prueba reduce el error al mínimo hasta el momento. El error promedio de E_{core} ha disminuido al 15.42 %, entrando dentro de lo que se considera aceptable en este estudio (20 %). El error máximo, en todos los casos se da cuando se trata de predecir $E_{core} = 1$ kPa. Se cree que esto puede deberse a la mayor variabilidad que sufre este parámetro, pues el rango que comprende va desde 1 kPa hasta 400 kPa, mientras que el rango de E_{placa} va desde 390 kPa hasta 1200 kPa. Por este motivo, se cree que podría ser interesante calcular una nueva base de datos que minimice el efecto de esta variabilidad. Esto es, reducir el rango de E_{core} a 1 kPa-100 kPa (como se ha explicado en la sección 3.1, y generar un vector que siga una distribución logarítmica, de manera que se obtengan muchas más configuraciones de casos que contemplen valores entre 1 kPa y 20 kPa, que es una zona cercana a los valores conflictivos. Esto podría mejorar significativamente la predicción. Además, 400 casos parecen ser muy pocos para que la herramienta reconozca los patrones internos en la base de datos y tenga capacidad de aprender para predecir valores de diferentes geometrías. Siguiendo esta idea, se realiza una prueba más antes de generar la nueva base de datos. Esta prueba consiste en intentar predecir E_{core} y E_{placa} pasando como entrada los datos de las deformaciones de una nueva geometría con grado de estenosis 60 % (Prueba 5).

Al realizar esta quinta prueba, se observa que la predicción es muy mala, revelando que la variable $E_{stenosis}(\%)$ es crucial para conseguir una buena predicción dada su influencia en la deformación. Es en este punto donde se decide considerarla en todos los modelos, lo que implica crear 8 geometrías extra que abarquen el rango 40 % - 80 % como ya se adelantaba en la sección 2.2.

Una vez se dispone de las nuevas geometrías, se cambia la metodología de trabajo, generando todos los casos de estudio utilizando la distribución logarítmica que se ha comentado anteriormente. De este modo, se cuenta con una nueva base de datos que cuenta con 26 mil datos diferentes. Con estos datos e incluyendo el grado de estenosis como una variable más de entrada, se lanza la sexta prueba. Los

resultados de esta prueba muestran una mejoría considerable. Sin embargo, podría no parecer lo mismo si se observa el error máximo obtenido: 151 %. Dado que se ha alcanzado una estructura de la red adecuada, se considera relevante reflejar con mayor detalle la dispersión de los resultados de la predicción (ver figura 4.2). Esta gráfica muestra que la mayoría de los datos predichos tienen un error menor al 20 %. Además, se ha observado que los datos atípicos que se corresponden con errores por encima de este porcentaje, se corresponden con los valores cercanos a 1 kPa (por ejemplo: valor predicho 2.07 kPa, valor real 1.07 kPa). Para conocer el impacto real de asumir este error, se ha comprobado su influencia en la predicción de la vulnerabilidad de la placa mediante un análisis de elementos finitos en ABAQUS, y se ha observado que las tensiones máximas en la zona crítica resultantes de introducir un material con 2.07 kPa o de 1.07 kPa son prácticamente las mismas, siendo la diferencia máxima del 0,56 %. Así que este error se considera asumible.

	Nº Capas	Nº neuronas/capa	Func. Activación	Tamaño lote
Prueba 1	4	50, 25, 10, 2	ReLU	16
Prueba 2.a	4	50, 25, 10, 2	Elu	16
Prueba 2.b	4	50, 25, 10, 2	Selu	16
Prueba 2.c	4	50, 25, 10, 2	Softplus	16
Prueba 3	8	250, 150, 100, 50, 40, 30, 10, 2	Softplus	16
Prueba 4	8	250, 150, 100, 50, 40, 30, 10, 2	Softplus	8
Prueba 5	8	250, 150, 100, 50, 40, 30, 10, 2	Softplus	8
Prueba 6	8	250, 150, 100, 50, 40, 30, 10, 2	Softplus	8

Tabla 4.1: Estructura de las diferentes Redes Neuronales Artificiales analizadas.

	E_{core}		
	Error medio (%)	Error máximo (%)	Error mínimo (%)
Prueba 1	100	100	100
Prueba 2.a	1561	25246	1.13
Prueba 2.b	1672	25465	0.7539
Prueba 2.c	1578	25952	1.98
Prueba 3	55.747	275.8	0.131
Prueba 4	15.428	99.99	0.788
Prueba 5	141.18	3546	0.68
Prueba 6	10.26	151.08	0.0035

Tabla 4.2: Resultados de la predicción de E_{core} de las diferentes Redes Neuronales Artificiales analizadas.

	E_{placa}		
	Error medio (%)	Error máximo (%)	Error mínimo (%)
Prueba 1	3.5	5.96	0.11
Prueba 2.a	4.85	14.14	0.23
Prueba 2.b	6.95	18.11	0.084
Prueba 2.c	4.3	10.57	0.199
Prueba 3	2.9	59.12	0.07
Prueba 4	1.08	3.4	0.15
Prueba 5	11.8	24.28	0.975
Prueba 6	0.196	1.38	0.00015

Tabla 4.3: Resultados de la predicción de E_{placa} de las diferentes Redes Neuronales Artificiales analizadas.

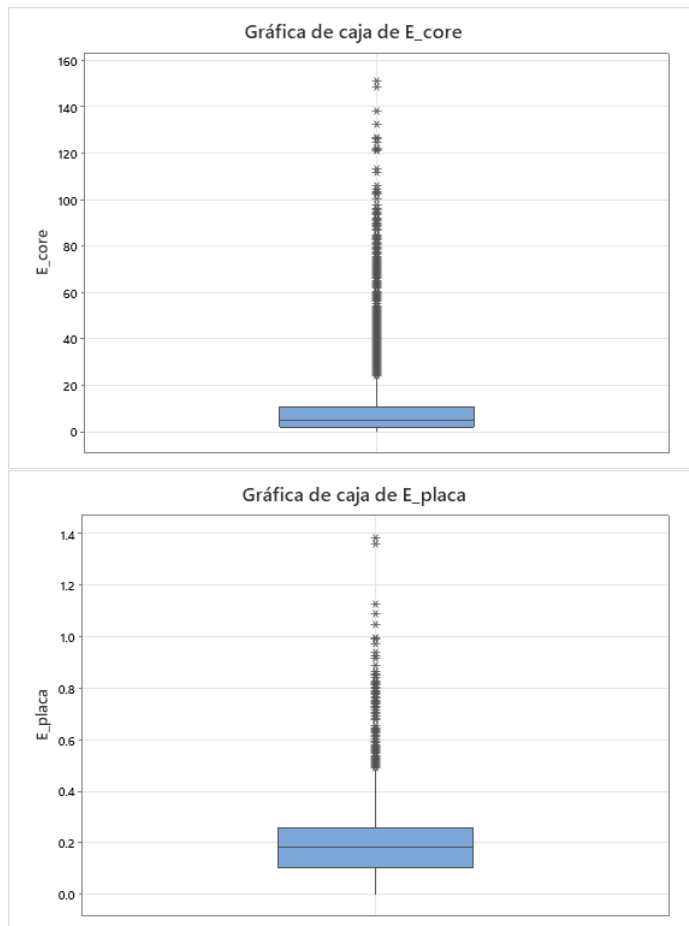


Figura 4.2: Gráfica de caja de las variables E_{core} y E_{placa} .

4.4. Validación con geometrías reales

Una vez seleccionada la red, entrenada y testeada con los datos procedentes de geometrías idealizadas con diferentes grados de estenosis, se procede a validar la RNA con geometrías de arterias reales obtenidas a partir de IVUS. Partiendo de la Placa nº1 presentada por [12], la cual cuenta con una capa fibrótica de $300\ \mu m$ y un área de lípido considerablemente inferior a la utilizada en los modelos ideales desarrollados en el presente proyecto, y se realizan dos modificaciones adicionales para estudiar qué parámetros geométricos que no han sido considerados hasta el momento podrían resultar de interés.

- Geometría realista 1: espesor de la capa fibrótica a $65\ \mu m$ y área del núcleo lipídico igual al de la geometría idealizada
- Geometría realista 2: se modifica únicamente el espesor de la capa fibrótica a $65\ \mu m$
- Geometría realista 3: no se modifica nada, por lo que el espesor de la capa fibrótica se mantiene en $300\ \mu m$ y el área del núcleo lipídico es 2 veces más pequeña que en geometría idealizada

Se modelan estas geometrías en ABAQUS y se calculan las deformaciones $\varepsilon_{core-cuad}$, $\varepsilon_{cap-cuad}$ y $\varepsilon_{theta-cuad}$. Estas variables, junto con el grado de estenosis, serán las nuevas entradas de la RNA previamente entrenada. Tras analizar los resultados de la predicción, se encuentra que no son suficientemente precisos (tabla 4.4). Esto puede ser debido a dos factores: la red está sobreentrenada, ya que es capaz de tener errores bajos para los datos de entrenamiento pero el error aumenta para nuevos datos, y la otra que la geometría influye en el resultado de la predicción, y que variables geométricas como el espesor de la capa fibrótica o el tamaño del lípido pueden afectar a la predicción

La hipótesis del sobreentrenamiento, puede ser comprobada y solventada simplemente añadiendo capas de **Dropout** a la red. Tras esta modificación, los resultados que se muestran en la tabla 4.5, revelan que la predicción en la geometría 1 es muy buena, mientras que en el resto no tanto. Esto reafirma la hipótesis de que los factores geométricos como el espesor de la capa fibrótica o el área del lípido podrían ser muy influyentes, dado que la predicción empeora para las geometrías 2 y 3. La única manera de solventar este inconveniente es generando una base de datos que tenga en cuenta más parámetros geométricos como los mencionados, e introducirlos como parámetros de entrada en la red para ayudar a mejorar la predicción.

Valores Entrada				
Geometría	$\varepsilon_{core-cuad}$	$\varepsilon_{cap-cuad}$	$\varepsilon_{theta-cuad}$	$E_{stenosis}(\%)$
1	0.000133	0.000393	0.000192	65.1
2	0.0000274	0.000276	0.000122	65.1
3	0.00000143	0.00886	0.000131	65.1
	Predicción		Real	
	$E_{core}(kPa)$	$E_{placa}(kPa)$	$E_{core}(kPa)$	$E_{placa}(kPa)$
1	0.0219	840	11.1	601
2	46.26	851	11.1	601
3	0.0000132	1246	11.1	601

Tabla 4.4: Resultados de la validación de la RNA prediciendo valores de E_{core} y E_{placa} correspondientes a nuevas geometrías no idealizadas.

Valores Entrada				
Geometría	$\varepsilon_{core-cuad}$	$\varepsilon_{cap-cuad}$	$\varepsilon_{theta-cuad}$	$E_{stenosis}(\%)$
1	0.000133	0.000393	0.000192	65.1
2	0.0000274	0.000276	0.000122	65.1
3	0.00000143	0.00886	0.000131	65.1
	Predicción		Real	
	$E_{core}(kPa)$	$E_{placa}(kPa)$	$E_{core}(kPa)$	$E_{placa}(kPa)$
1	11.2	677	11.1	601
2	88.96	717.34	11.1	601
3	0	1246	11.1	601

Tabla 4.5: Resultados de la validación de la RNA prediciendo valores de E_{core} y E_{placa} correspondientes a nuevas geometrías no idealizadas.

Capítulo 5

Conclusiones

Una vez se han analizado todos los resultados, las conclusiones que se han extraído son las siguientes:

- Se ha conseguido desarrollar una herramienta capaz de predecir con un error asumible geometrías que tengan unos valores similares entre sí de área del núcleo lipídico y espesor de la capa fibrótica.
- Las variables $\varepsilon_{core-cuad}$, $\varepsilon_{cap-cuad}$, $\varepsilon_{theta-cuad}$ y $Estenosis(\%)$ son muy importantes para la predicción de las propiedades elásticas de los diferentes materiales que componen la arteria coronaria aterosclerótica, pero podrían no ser suficientes, ya que los errores cometidos al predecir las propiedades elásticas de las geometrías reales 2 y 3 de validación, son muy elevados.
- Dado que las arterias reales pueden variar geométricamente en función del paciente, y que la incorporación del grado de estenosis como variable de entrada ha mejorado considerablemente la capacidad de predicción de la RNA, se cree necesario introducir más parámetros geométricos de entrada para mejorar su capacidad de predicción. Sin embargo, esto excede el alcance del proyecto.
- La generación de una RNA para predecir las propiedades elásticas de la arteria coronaria es viable siempre que se cuente con una base de datos suficientemente amplia que contenga mayor cantidad de datos en las configuraciones que comprenden el rango 1 kPa - 20 kPa, puesto que es donde mayor error se ha observado en la predicción.
- Por último, los indicios de que esta metodología es factible permite pensar que sería posible desarrollar una herramienta que permita predecir dichas variables para cada paciente de forma rápida.

5.1. Limitaciones

- En este proyecto no se ha contado con mapas de deformaciones reales provenientes de imágenes IVUS reales, sino que se ha trabajado con una única geometría de literatura que se ha modelado y modificado para obtener un mapa de deformaciones.
- Al final del proyecto, tras poder contar con una geometría real, se ha intentado validar la RNA, y se ha observado que la base de datos podría ser escasa dada la variabilidad de geometrías que se pueden encontrar en la vida real.
- Las simplificaciones del trabajo como que se hayan creado modelos en 2D o que se hayan utilizado modelos constitutivos simples como el Neo-Hookeano puede provocar que los patrones a reconocer por la RNA sean más simples de lo que son en la realidad.
- El error en las predicciones puede ser muy elevado si no se cuenta con una estructura de RNA adecuada, y dado que el método establecido para seleccionar la estructura es prueba-error, podrían existir otras estructuras que funcionen mejor y no se hayan utilizado en este proyecto.

5.2. Trabajo futuro

Este proyecto ha creado un camino a seguir para mejorar en la predicción de las propiedades elásticas de la placa de ateroma en paciente específico, y marca las pautas a analizar para mejorar las predicciones en cualquier paciente. Ya se ha dado el primer paso al crear diferentes geometrías con distintos grados de estenosis, pero será necesario contar con una base de datos más amplia que contemple varios aspectos geométricos como: diferentes espesores de capa fibrótica, distintas áreas del núcleo lipídico u otras morfologías de núcleos con mayor ángulo entre codos. Por último, podría ser interesante analizar otras técnicas de *Machine Learning* más allá de las Redes Neuronales Artificiales.

Bibliografía

- [1] Gregory A Roth, Degu Abate, Kalkidan Hassen Abate, Solomon M Abay, Cristiana Abbafati, Nooshin Abbasi, Hedayat Abbastabar, Foad Abd-Allah, Jemal Abdela, Ahmed Abdelalim, et al. Global, regional, and national age-sex-specific mortality for 282 causes of death in 195 countries and territories, 1980–2017: a systematic analysis for the global burden of disease study 2017. *The Lancet*, 392(10159):1736–1788, 2018.
- [2] American Heart Association et al. Heart and stroke statistical update. dallas, texas: American heart association, 2000, 2001.
- [3] World Health Organization. Cardiovascular diseases (CVDs). [https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)).
- [4] Fernando Villar, JR Banegas, J deM Donado, and F Rodríguez Artalejo. Las enfermedades cardiovasculares y sus factores de riesgo en españa: hechos y cifras. *Informe Sea*, 2007.
- [5] Johannes A Schaar, James E Muller, Erling Falk, Renu Virmani, Valentin Fuster, Patrick W Serruys, Antonio Colombo, Christodoulos Stefanadis, S Ward Casscells, Pedro R Moreno, et al. Terminology for high-risk and vulnerable coronary artery plaques. *European Heart Journal*, 25(12):1077–1082, 2004.
- [6] J. Ohayon, G. Finet, A. M. Gharib, D. A. Herzka, P. Tracqui, J. Heroux, G. Rioufol, M. S. Kotys, A. Elagha, and R. I. Pettigrew. Necrotic core thickness and positive arterial remodeling index: emergent biomechanical factors for evaluating the risk of plaque rupture. *Am J Physiol Heart Circ Physiol*, 295:H717–H727, 2008.
- [7] M Cilla, E Peña, and MA Martínez. 3D computational parametric analysis of eccentric atheroma plaque: influence of axial and circumferential residual stresses. *Biomechanics and Modeling in Mechanobiology*, 11(7):1001–1013, 2012.

- [8] F. J. Gijssen and F. Migliavacca. Plaque mechanics. *Journal of Biomechanics*, 47(4):763–764, 2014.
- [9] Min Cheol Whang. *Stress analysis of the diseased arterial cross-section*. PhD thesis, Georgia Institute of Technology, 1990.
- [10] Roch L Maurice, Jérémie Fromageau, Élisabeth Brusseau, Gérard Finet, Gilles Rioufol, and Guy Cloutier. On the potential of the lagrangian estimator for endovascular ultrasound elastography: in vivo human coronary artery study. *Ultrasound in Medicine & Biology*, 33(8):1199–1205, 2007.
- [11] Troels Thim. Human-like atherosclerosis in minipigs: a new model for detection and treatment of vulnerable plaques. *Danish Medical Bulletin*, 57(7):B4161, 2010.
- [12] Simon Le Floc’h, Jacques Ohayon, Philippe Tracqui, Gérard Finet, Ahmed M Gharib, Roch L Maurice, Guy Cloutier, and Roderic I Pettigrew. Vulnerable atherosclerotic plaque elasticity reconstruction based on a segmentation-driven optimization procedure using strain measurements: theoretical framework. *IEEE Transactions on Medical Imaging*, 28(7):1126–1137, 2009.
- [13] Antoine Tacheau, Simon Le Floc’h, Gérard Finet, Marvin M Doyley, Roderic I Pettigrew, Guy Cloutier, and Jacques Ohayon. The imaging modulography technique revisited for high-definition intravascular ultrasound: Theoretical framework. *Ultrasound in Medicine & Biology*, 42(3):727–741, 2016.
- [14] Myriam Cilla, Javier Martinez, Estefania Pena, and Miguel Ángel Martínez. Machine learning techniques as a helpful tool toward determination of plaque vulnerability. *IEEE Transactions on Biomedical Engineering*, 59(4):1155–1161, 2012.
- [15] Estefanía Peña, Myriam Cilla, Álvaro T. Latorre, Miguel A. Martínez, Armida Gómez, Roderic I. Pettigrew, Gérard Finet, and Jacques Ohayon. Chapter 16 - emergent biomechanical factors predicting vulnerable coronary atherosclerotic plaque rupture. In Jacques Ohayon, Gerard Finet, and Roderic Ivan Pettigrew, editors, *Biomechanics of Coronary Atherosclerotic Plaque*, volume 4 of *Biomechanics of Living Organs*, pages 361–380. Academic Press, 2021.
- [16] Ali C Akyildiz, Lambert Speelman, Bas van Velzen, Raoul RF Stevens, Antonius FW Van Der Steen, Wouter Huberts, and Frank JH Gijssen. Intima heterogeneity in stress assessment of atherosclerotic plaques. *Interface Focus*, 8(1):20170008, 2018.

- [17] Gérard Finet, Jacques Ohayon, and Gilles Rioufol. Biomechanical interaction between cap thickness, lipid core composition and blood pressure in vulnerable coronary plaque: impact on stability or instability. *Coronary Artery Disease*, 15(1):13–20, 2004.
- [18] George C Cheng, Howard M Loree, Roger D Kamm, Michael C Fishbein, and Richard T Lee. Distribution of circumferential stress in ruptured and stable atherosclerotic lesions. a structural analysis with histopathological correlation. *Circulation*, 87(4):1179–1187, 1993.
- [19] Peter R Hoskins, Peter R Hoskins, Patricia V Lawford, Patricia V Lawford, Barry J Doyle, and Barry J Doyle. *Cardiovascular Biomechanics*. Springer, 2017.
- [20] Armida Gómez, Antoine Tacheau, Gerard Finet, Manuel Lagache, Jean-Louis Martiel, Simon Le Floc’h, Saami K Yazdani, Alex Elias-Zuñiga, Roderic I Pettigrew, Guy Cloutier, et al. Intraluminal ultrasonic palpation imaging technique revisited for anisotropic characterization of healthy and atherosclerotic coronary arteries: A feasibility study. *Ultrasound in Medicine & Biology*, 45(1):35–49, 2019.
- [21] Song Jiangping, Zheng Zhe, Wang Wei, Song Yunhu, Huang Jie, Wang Hongyue, Zhao Hong, and Hu Shengshou. Assessment of coronary artery stenosis by coronary angiography: a head-to-head comparison with pathological coronary artery anatomy. *Circulation: Cardiovascular Interventions*, 6(3):262–268, 2013.
- [22] Gerhard A Holzapfel, Thomas C Gasser, and Ray W Ogden. A new constitutive framework for arterial wall mechanics and a comparative study of material models. *Journal of Elasticity*, 61(1):1–48, 2000.
- [23] T Christian Gasser, Ray W Ogden, and Gerhard A Holzapfel. Hyperelastic modelling of arterial layers with distributed collagen fibre orientations. *Journal of the Royal Society Interface*, 3(6):15–35, 2006.
- [24] Jorge Mendez Gonzalez. Stepwise regresión. https://rpubs.com/jorge_mendez/609253#:~:text=La%20regresi%C3%B3n%20paso%20a%20paso,modelos%20con%20cientos%20de%20variables., 2019.
- [25] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian

- Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.
- [26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [27] Luis Velasco. Optimizadores en redes neuronales profundas: un enfoque práctico. <https://velascoluis.medium.com/optimizadores-en-redes-neuronales-profundas-un-enfoque-pr%C3%A1ctico-819b39a3eb5>, 2020.
- [28] Guillermo Herrero Peña. Análisis mecánico de placas de ateroma en fase de crecimiento positivo: Estudio de la influencia de la longitud de la placa. *Universidad de Zaragoza*, 2015.
- [29] Sagar Sharma and Simone Sharma. Activation functions in neural networks. *Towards Data Science*, 6(12):310–316, 2017.
- [30] Javier Luna Gonzalez. Tipos de aprendizaje automático. <https://medium.com/soldai/tipos-de-aprendizaje-autom%C3%A1tico-6413e3c615e2>, 2018.

Anexo A

Geometrías

Conocida la geometría de la arteria coronaria aterosclerótica (figura 2.1), y la expresión matemática de la evolución del grado de estenosis (ecuación 2.1) [6] que depende del área de la placa y del área del lumen, se pueden crear diferentes geometrías modificando únicamente estos parámetros sobre la geometría original. De este modo, y debido a las necesidades del proyecto, se generan 9 geometrías diferentes que cubren el rango 40 % - 80 % de estenosis. Para modificar estos parámetros se han tomado los valores numéricos de los parámetros geométricos que forman parte de la arteria basándose en el trabajo previamente realizado por Guillermo Herrero en su Trabajo Final de Grado [28] (figura A.1).

Degree of stenosis %	Lumen area (mm ²)	Plaque area (mm ²)	Lumen radius (mm)	Plaque radius (mm)	Media radius (mm)	Adventitia radius (mm)	Remodeling index	Plaque burden %
0,000	13,125	0,000	2,044	2,044	2,344	2,644	1,000	23,959
5,000	12,843	0,676	2,022	2,074	2,371	2,668	1,023	27,253
10,000	12,544	1,394	1,998	2,106	2,399	2,692	1,047	30,594
15,000	12,225	2,157	1,973	2,140	2,428	2,719	1,073	33,982
20,000	11,886	2,971	1,945	2,175	2,459	2,746	1,100	37,419
25,000	11,523	3,841	1,915	2,211	2,491	2,775	1,130	40,906
30,000	11,135	4,772	1,883	2,250	2,526	2,806	1,161	44,444
35,000	10,718	5,771	1,847	2,291	2,562	2,839	1,195	48,033
40,000	10,270	6,847	1,808	2,334	2,601	2,874	1,231	51,676
45,000	9,786	8,007	1,765	2,380	2,642	2,911	1,270	55,373
50,000	9,263	9,263	1,717	2,428	2,686	2,951	1,313	59,125
55,000	8,694	10,626	1,664	2,480	2,732	2,994	1,359	62,934
60,000	8,074	12,112	1,603	2,535	2,782	3,039	1,409	66,801
65,000	7,397	13,737	1,534	2,594	2,836	3,089	1,464	70,728
70,000	6,652	15,522	1,455	2,657	2,894	3,142	1,524	74,716
75,000	5,831	17,492	1,362	2,725	2,956	3,199	1,591	78,765
80,000	4,919	19,678	1,251	2,798	3,024	3,262	1,665	82,879
85,000	3,903	22,116	1,115	2,878	3,098	3,331	1,747	87,057
90,000	2,761	24,853	0,938	2,965	3,179	3,406	1,839	91,303
95,000	1,471	27,947	0,684	3,060	3,268	3,490	1,944	95,616
100,000	0,000	31,475	0,000	3,165	3,367	3,582	2,063	100,000

Figura A.1: Tabla con el cálculo de los datos de cada sección en función del grado de estenosis.

Así, se han generado las 9 geometrías, todas ellas con un espesor de la capa fibrótica de 65 μm y el mismo área de lípido:

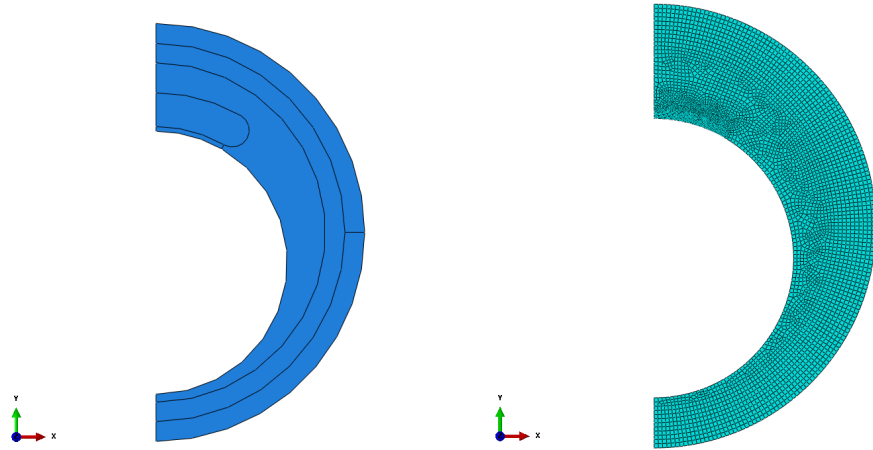


Figura A.2: Geometría y mallado correspondiente a Estenosis 40 %.

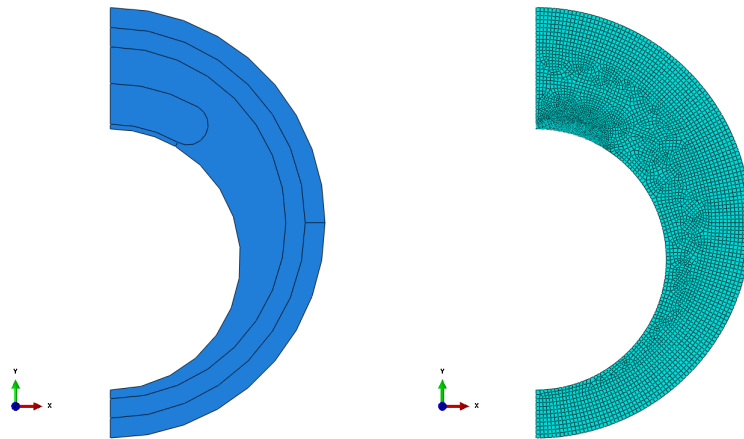


Figura A.3: Geometría y mallado correspondiente a Estenosis 45 %.

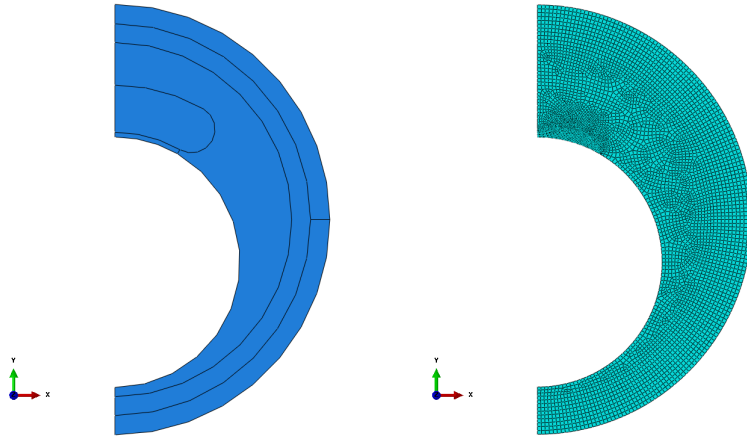


Figura A.4: Geometría y mallado correspondiente a Estenosis 50 %.

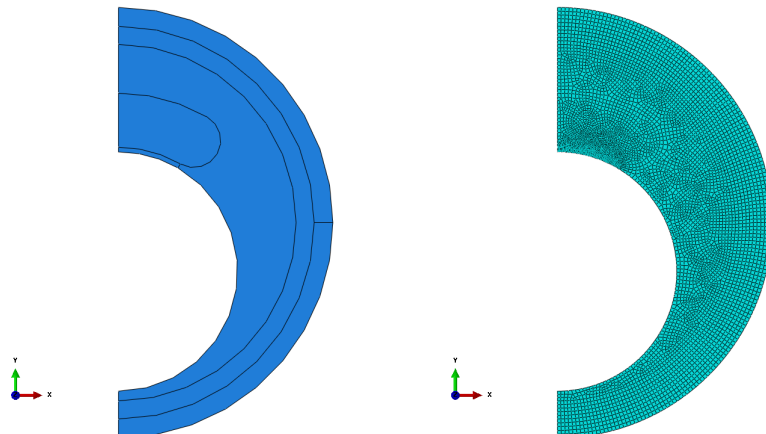


Figura A.5: Geometría y mallado correspondiente a Estenosis 55 %.

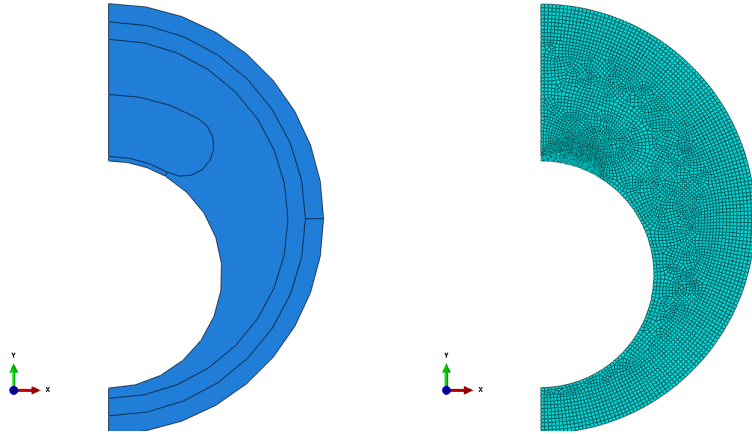


Figura A.6: Geometría y mallado correspondiente a Estenosis 60 %.

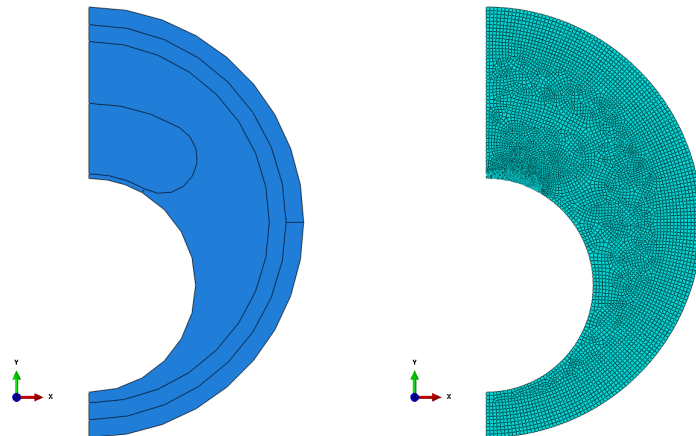


Figura A.7: Geometría y mallado correspondiente a Estenosis 65 %.

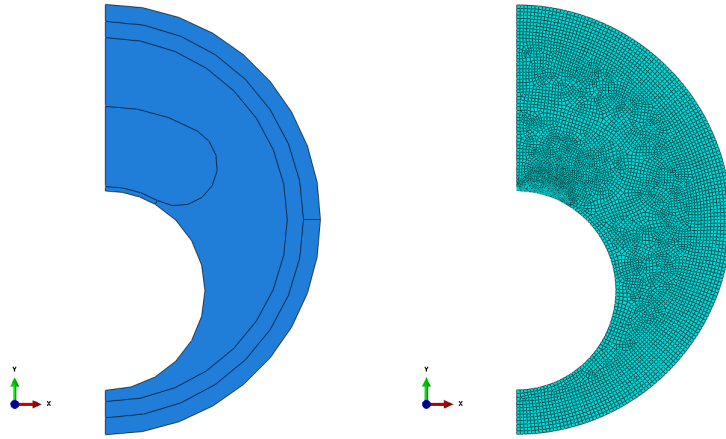


Figura A.8: Geometría y mallado correspondiente a Estenosis 70 %.

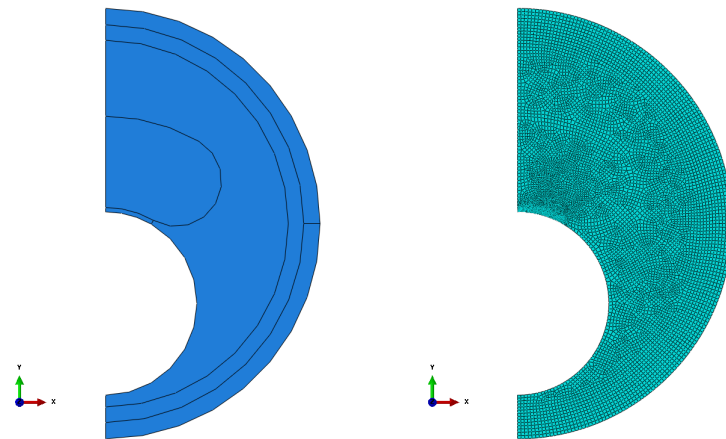


Figura A.9: Geometría y mallado correspondiente a Estenosis 75 %.

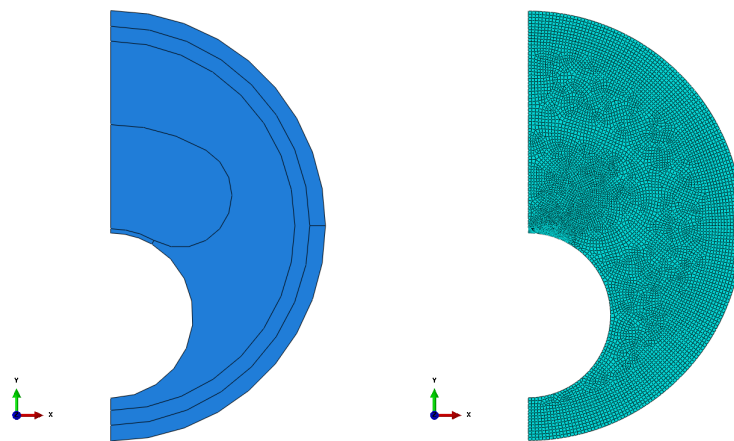


Figura A.10: Geometría y mallado correspondiente a Estenosis 80 %.

Anexo B

Generación de la base de datos a partir de modelos de elementos finitos

Este Anexo recoge los comandos utilizados con el código completo para la generación de la base de datos mediante el uso del módulo de *Estudios Paramétricos* de ABAQUS. Un estudio paramétrico en ABAQUS está compuesto de las siguientes partes:

- El estudio se crea con el comando `ParStudy` e indicando cuales son los parámetros del estudio: E_{core} y E_{placa} .
- Después es necesario definir si los parámetros van a tomar valores continuos o discretos, así como el dominio de estos. En este punto se ha decidido establecer valores discretos de E_{core} , ya que la manera en la que ABAQUS interpreta las líneas de código en lenguaje Python, no permite crear una distribución logarítmica de los inputs, sino que solo trabaja con vectores linealmente espaciados. Sin embargo, es posible crear la distribución logarítmica deseada fuera de ABAQUS y después pasarle manualmente esos datos como entradas. El motivo de necesitar una distribución discreta es que, como se ha visto en la sección 3, es necesario contar con una mayor cantidad de información en los valores bajos de E_{core} (Rango entre 1 y 20 kPa) para mejorar la capacidad de predicción de la red neuronal. Por tanto, se optó por que los diferentes valores que podía tomar la variable E_{core} , siguieran una distribución logarítmica (figura B.1). Una vez se ha definido el dominio, se introducen mediante el comando `sample values` los valores concretos que ha de tomar cada variable. En el caso de E_{placa} , no se hace necesario que tome valores con una distribución logarítmica, y por tanto se ha definido

como una variable continua. Su dominio comprende el rango 390 – 1200 kPa y se han generado únicamente 10 valores.

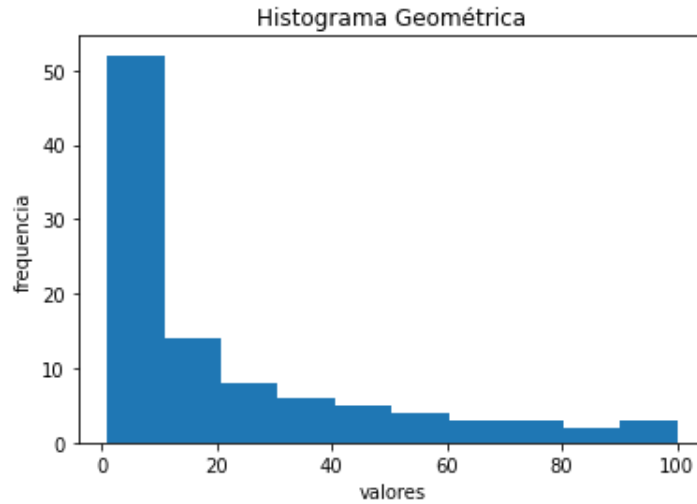


Figura B.1: Distribución logarítmica de los valores que componen el estudio paramétrico.

Tras definir los valores que va a tomar cada variable, es necesario crear cada caso de estudio combinando cada uno de ellos. Para esto se utiliza el comando `combine` (MESH) (figura B.2).

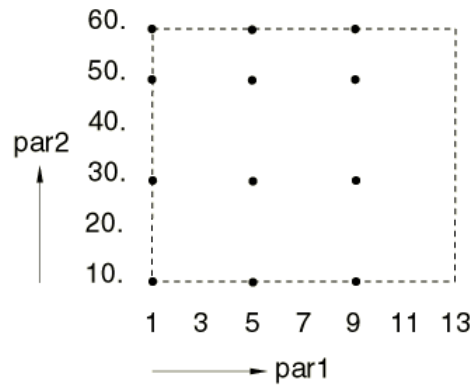


Figura B.2: Combinación de casos en el estudio paramétrico.

El siguiente paso es generar las configuraciones de estudio el comando `generate`. Este comando tomará como plantilla el fichero `.inp` generado y modificado previamente, e irá sustituyendo cada una de las combinaciones de valores que pueden

tomar E_{core} y E_{placa} en la variable parametrizada correspondiente (C_{core}, C_{placa}). En otras palabras, las variables parametrizadas C_{core} y C_{placa} se sustituyen por los diferentes valores que van tomando $E_{core}/6$ y $E_{placa}/6$ respectivamente a lo largo del estudio. Una vez generados todos los casos de interés, se ejecutan con el comando `execute`. Después, se indican cuáles han de ser los datos de salida que se obtendrán a partir de los ficheros `.ODB`, que son los ficheros que recogen los resultados del análisis. Por último, se ordenan todos los resultados en una tabla con formato de fichero de texto con los comandos `gather` y `report`. En las siguientes líneas se muestra el código utilizado para llevar a cabo estas acciones.

```
EstudioParametricoV1=ParStudy (par=('Ecore','Eplaca'),
name='EstudioParametricoV1')

EstudioParametricoV1.define(DISCRETE, par='Ecore',
domain=(1,100))

EstudioParametricoV1.sample(VALUES, par='Ecore', values=(
1., 1.01396764, 1.02813038, 1.04249094,
1.05705208, 1.07181661, 1.08678736, 1.10196722,
1.1173591, 1.13296598, 1.14879084, 1.16483674,
1.18110676, 1.19760404, 1.21433175, 1.2312931,
1.24849136, 1.26592984, 1.2836119, 1.30154093,
1.31972039, 1.33815378, 1.35684463, 1.37579655,
1.39501319, 1.41449823, 1.43425544, 1.45428861,
1.47460159, 1.4951983, 1.5160827, 1.5372588,
1.55873068, 1.58050248, 1.60257837, 1.62496261,
1.64765951, 1.67067343, 1.6940088, 1.71767011,
1.74166192, 1.76598883, 1.79065553, 1.81566677,
1.84102735, 1.86674217, 1.89281615, 1.91925433,
1.94606179, 1.97324369, 2.00080525, 2.02875179,
2.05708867, 2.08582135, 2.11495536, 2.1444963,
2.17444986, 2.2048218, 2.23561796, 2.26684428,
2.29850675, 2.33061147, 2.36316462, 2.39617246,
2.42964134, 2.4635777, 2.49798808, 2.53287908,
2.56825744, 2.60412994, 2.6405035, 2.67738511,
2.71478187, 2.75270097, 2.79114972, 2.8301355,
2.86966582, 2.90974829, 2.95039062, 2.99160062,
3.03338623, 3.07575549, 3.11871654, 3.16227766,
3.20644723, 3.25123374, 3.29664581, 3.34269218,
3.38938171, 3.43672339, 3.48472631, 3.53339973,
3.58275299, 3.63279561, 3.6835372, 3.73498753,
3.78715651, 3.84005416, 3.89369066, 3.94807635,
4.00322167, 4.05913724, 4.11583382, 4.17332232,
4.23161379, 4.29071946, 4.3506507, 4.41141904,
4.47303617, 4.53551394, 4.59886438, 4.66309968,
4.72823219, 4.79427445, 4.86123916, 4.92913922,
```

4.99798767 ,	5.06779778 ,	5.13858297 ,	5.21035687 ,
5.28313327 ,	5.35692619 ,	5.43174982 ,	5.50761857 ,
5.58454702 ,	5.66254998 ,	5.74164246 ,	5.82183967 ,
5.90315705 ,	5.98561024 ,	6.06921511 ,	6.15398774 ,
6.23994444 ,	6.32710176 ,	6.41547646 ,	6.50508554 ,
6.59594626 ,	6.68807608 ,	6.78149274 ,	6.87621421 ,
6.97225872 ,	7.06964474 ,	7.16839101 ,	7.26851654 ,
7.37004059 ,	7.47298268 ,	7.57736264 ,	7.68320054 ,
7.79051674 ,	7.8993319 ,	8.00966695 ,	8.12154312 ,
8.23498193 ,	8.35000522 ,	8.46663511 ,	8.58489405 ,
8.70480479 ,	8.82639039 ,	8.94967427 ,	9.07468012 ,
9.20143202 ,	9.32995433 ,	9.46027181 ,	9.59240951 ,
9.72639286 ,	9.86224764 ,	10. ,	10.13967643 ,
10.28130381 ,	10.4249094 ,	10.57052081 ,	10.71816607 ,
10.86787359 ,	11.01967217 ,	11.17359102 ,	11.32965975 ,
11.4879084 ,	11.6483674 ,	11.81106764 ,	11.97604042 ,
12.14331748 ,	12.312931 ,	12.48491363 ,	12.65929845 ,
12.83611901 ,	13.01540934 ,	13.19720393 ,	13.38153777 ,
13.56844631 ,	13.75796553 ,	13.95013188 ,	14.14498234 ,
14.34255441 ,	14.54288609 ,	14.74601593 ,	14.95198302 ,
15.16082698 ,	15.372588 ,	15.58730683 ,	15.80502477 ,
16.02578371 ,	16.24962614 ,	16.47659512 ,	16.70673432 ,
16.94008802 ,	17.17670113 ,	17.41661916 ,	17.65988828 ,
17.9065553 ,	18.15666767 ,	18.41027353 ,	18.66742166 ,
18.92816154 ,	19.19254335 ,	19.46061794 ,	19.73243691 ,
20.00805255 ,	20.28751788 ,	20.57088669 ,	20.8582135 ,
21.14955358 ,	21.444963 ,	21.74449859 ,	22.04821798 ,
22.35617962 ,	22.66844276 ,	22.98506748 ,	23.3061147 ,
23.63164619 ,	23.9617246 ,	24.29641341 ,	24.63577705 ,
24.97988079 ,	25.32879085 ,	25.68257436 ,	26.04129939 ,
26.40503497 ,	26.77385108 ,	27.14781867 ,	27.52700972 ,
27.91149716 ,	28.301355 ,	28.69665822 ,	29.0974829 ,
29.50390616 ,	29.91600619 ,	30.33386229 ,	30.75755486 ,
31.1871654 ,	31.6227766 ,	32.06447226 ,	32.51233737 ,
32.96645809 ,	33.42692181 ,	33.89381713 ,	34.36723387 ,
34.84726313 ,	35.33399726 ,	35.82752993 ,	36.32795608 ,
36.83537201 ,	37.34987534 ,	37.87156507 ,	38.40054157 ,
38.93690664 ,	39.48076345 ,	40.03221667 ,	40.59137238 ,
41.15833819 ,	41.73322317 ,	42.31613793 ,	42.90719465 ,
43.50650703 ,	44.11419039 ,	44.73036166 ,	45.35513939 ,
45.98864379 ,	46.63099676 ,	47.28232188 ,	47.94274448 ,
48.61239162 ,	49.29139216 ,	49.97987674 ,	50.67797782 ,
51.38582973 ,	52.10356866 ,	52.83133271 ,	53.56926192 ,
54.31749825 ,	55.07618568 ,	55.84547019 ,	56.62549979 ,
57.41642456 ,	58.21839669 ,	59.03157048 ,	59.85610239 ,
60.69215106 ,	61.53987737 ,	62.39944442 ,	63.27101759 ,
64.15476458 ,	65.05085544 ,	65.95946257 ,	66.88076081 ,
67.81492741 ,	68.76214211 ,	69.72258718 ,	70.69644739 ,
71.68391014 ,	72.68516542 ,	73.70040587 ,	74.72982683 ,


```

75.77362639, 76.83200536, 77.90516739, 78.99331897,
80.09666946, 81.21543115, 82.34981931, 83.5000522 ,
84.66635113, 85.84894051, 87.04804787, 88.26390394,
89.49674265, 90.74680122, 92.01432015, 93.29954334,
94.60271807, 95.92409507, 97.2639286 , 98.62247644,
100.))

EstudioParametricoV1.define(CONTINUOUS, par='Eplaca',
domain=(390, 1200))

EstudioParametricoV1.sample(NUMBER, par='Eplaca', number=10)

EstudioParametricoV1.combine (MESH)

EstudioParametricoV1.generate(template='Job-1')

EstudioParametricoV1.execute (ALL)

EstudioParametricoV1.output(file=ODB, step=1)

EstudioParametricoV1.gather(results='COORD_nodo14',
variable='COORD',node=14,request=FIELD, step=1,
instance="Part-1-1")
EstudioParametricoV1.gather(results='COORD_nodo7',
variable='COORD',node=7,request=FIELD, step=1,
instance="Part-1-1")
EstudioParametricoV1.gather(results='COORD_nodo8',
variable='COORD',node=8,request=FIELD, step=1,
instance="Part-1-1")
EstudioParametricoV1.gather(results='COORD_nodo15',
variable='COORD',node=15,request=FIELD, step=1,
instance="Part-1-1")

EstudioParametricoV1.report(FILE, results=('COORD_nodo14',
'COORD_nodo7','COORD_nodo8','COORD_nodo15'),
file='ResultadosDataSet.psr')

```


Anexo C

Redes Neuronales

El término Inteligencia Artificial hace referencia a la ciencia que impulsa el desarrollo de dispositivos capaces de llevar a cabo tareas que normalmente requieren de inteligencia humana, como la percepción visual, el reconocimiento de palabras y de texto, la toma de decisiones o el reconocimiento de patrones. Dentro del campo de la Inteligencia Artificial se pueden encontrar diferentes áreas que corresponden a una amplia variedad de comportamientos inteligentes: la capacidad de moverse e interactuar con el entorno (robótica); capacidad de entender el lenguaje (*natural language processing*); la capacidad de poder hablar, donde se estudia tanto la conversión de voz a texto como la conversión de texto a voz. Todas estas capacidades conforman campos de estudio propios dentro de la Inteligencia Artificial, pero, sin embargo, la capacidad que más representa al ser humano como seres inteligentes es la capacidad de aprender. Este campo es el que se conoce como Aprendizaje Automático o *Machine Learning* (ML) y su objetivo principal es dotar a las máquinas de capacidad de aprendizaje tratando de generalizar el conocimiento a partir de un conjunto de experiencias. De este modo, la diferencia esencial entre los términos Inteligencia Artificial y *Machine Learning*, es que en el primero se programa el dispositivo para que realice una tarea, mientras que en el segundo el objetivo es que aprenda a realizar la tarea. Existen tres tipos de aprendizaje automático (anexo C): supervisado, no supervisado y por refuerzo.

Como se puede observar, dentro del *Machine Learning*, existen multitud de técnicas que sirven para cubrir diferentes tipos de aplicaciones. Sin embargo, la técnica más utilizada durante la última década han sido las redes neuronales. La característica diferenciadora de las redes neuronales es que, gracias a su estructura jerarquizada, la información se aprende por niveles, donde en las primeras capas se aprenden conceptos muy concretos (cómo es un ojo, una nariz o una boca), y en las capas posteriores se utiliza esa información para aprender conceptos más abstractos (raza de una persona, estado de ánimo, edad de una persona, etc). De este modo, a

medida que se añaden más capas, la información que se aprende es cada vez más abstracta. Lo interesante es que no hay un límite establecido para el número de capas que se pueden añadir a una red neuronal, por lo que la tendencia es añadir cada vez más y más capas, creando así algoritmos cada vez más complejos. Este incremento en el número de capas es lo que hace que estos algoritmos sean conocidos como algoritmos de aprendizaje profundo (*Deep Learning*). En este trabajo se ha entrenado una red neuronal artificial mediante aprendizaje supervisado utilizando la base de datos previamente creada, con el objetivo de desarrollar una herramienta que sea capaz de predecir las propiedades elásticas de la arteria coronaria con aterosclerosis, y relacionando éstas, en última instancia, con la vulnerabilidad de la placa.

C.1. Concepto de Red Neuronal Artificial

La complejidad de las redes neuronales artificiales emerge de la interacción de muchas partes más simples trabajando conjuntamente. Cada una de estas partes se llama neurona. Por tanto, la neurona es la unidad básica de procesamiento que compone una red neuronal. Una neurona artificial es un sistema que intenta imitar el comportamiento de una neurona biológica. De este modo, las neuronas artificiales tienen conexiones de entrada a través de las cuales reciben estímulos externos que se almacenan de forma numérica como valores de entrada (X_i). Con estos valores de entrada la neurona realizará un cálculo interno y generará un valor de salida (Y_i). Por tanto, una neurona se puede asemejar a una función matemática que utiliza todos los valores de entrada para realizar una suma ponderada (ecuación C.1). La ponderación de las entradas X_i viene dada por los pesos W_i que se le asignan a cada una de las conexiones de entrada en función de la intensidad con la que afecte cada una de estos parámetros de entradas a la salida. El parámetro b es conocido como sesgo (en inglés, *bias*) y permite ajustar el término independiente de la ecuación C.1. Dado que una red neuronal es un conjunto de neuronas interaccionado entre sí de forma secuencial, matemáticamente se puede considerar como una concatenación de regresiones lineales C.1.

$$y = \sum_{i=1}^n W_i X_i + b \quad (\text{C.1})$$

Se puede comprobar que la operación resultante de sumar múltiples operaciones lineales equivale a haber realizado únicamente una operación. Por esta razón, es necesario manipular cada una de las regresiones lineales de modo que den un resultado no lineal. Esto lo hace la función de activación [29] mediante la manipulación de los datos de salida, distorsionándolos de forma no lineal para poder encadenar

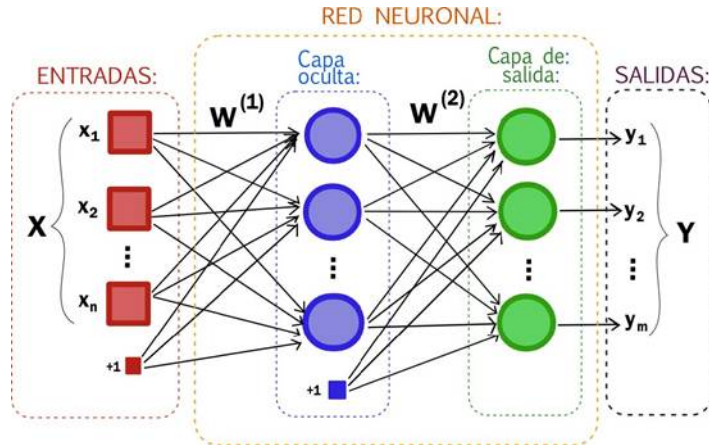


Figura C.1: Esquema del funcionamiento de una red neuronal artificial.

de manera efectiva una red de neuronas. Esta distorsión será diferente para cada función de activación. Por ejemplo, si se trabaja con la función de activación sigmoide (figura C.2), cuando el valor de entrada sea mucho mayor que el valor umbral, los valores de salida estarán saturados en uno. En caso contrario, estarán saturados en cero.

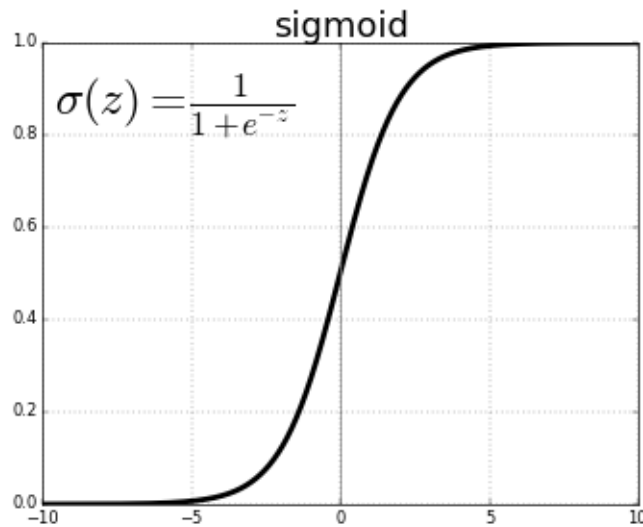


Figura C.2: Función de activación sigmoide

La función de activación más utilizada en ML en problemas de regresión es la función lineal rectificadora uniforme (ReLU). Como se observa en la figura C.3, esta

función toma valor cero para valores negativos y tiene un comportamiento lineal para valores positivos. Sin embargo, dado que esta función no es derivable en cero, y en este trabajo se manipulan valores muy cercanos al cero, puede ser problemática. Por ese motivo, la función de activación seleccionada ha sido la Softplus (ecuación C.2, figura C.3), que es una versión suavizada de la ReLU, y sí es derivable en cero.

$$\Phi(x) = \ln(1 + e^x) \quad (\text{C.2})$$

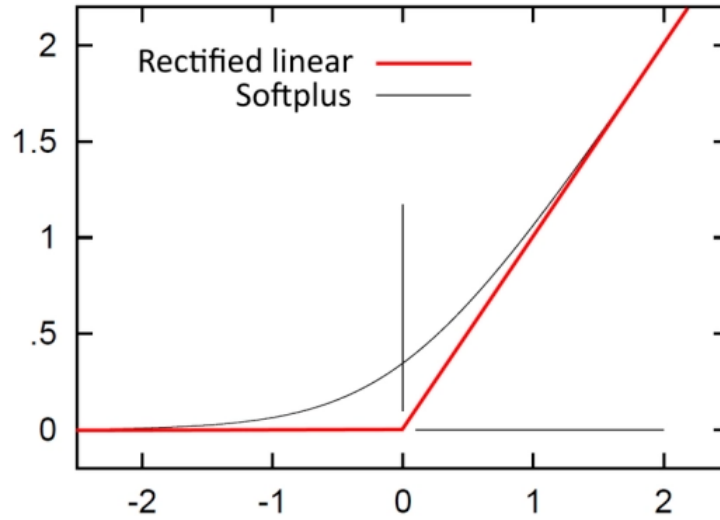


Figura C.3: Funciones de activación ReLU y Softplus

Para que una red neuronal sea capaz de predecir correctamente, los pesos (W_i) se ajustan automáticamente tratando de minimizar la función del error. Una de las funciones de error más utilizadas es la del error cuadrático medio (ecuación C.3).

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2 \quad (\text{C.3})$$

La función de coste se minimiza numéricamente mediante el método del descenso del gradiente. Sin embargo, cuando se trabaja con funciones de coste es común encontrar funciones no convexas (figura C.4) donde pueden existir múltiples puntos de pendiente nula donde la derivada es cero como máximos o mínimos locales y absolutos.

Para solucionar este inconveniente, el método de descenso del gradiente observa en cada iteración cuál es la dirección con mayor pendiente, es decir, aquella dirección

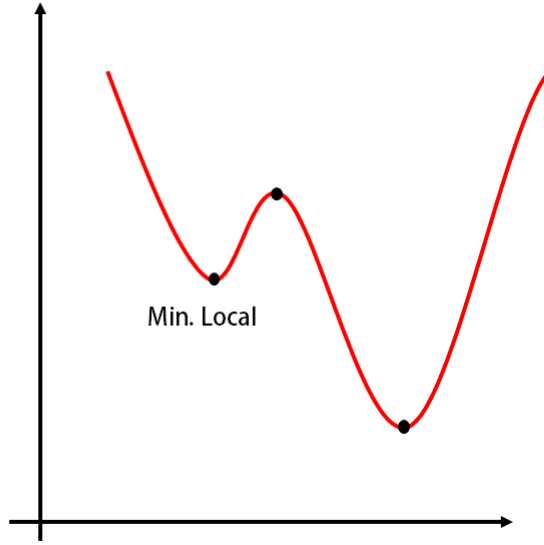


Figura C.4: Función no convexa con puntos de pendiente nula

que cambia con mayor rapidez. Traducido a las matemáticas, este método evalúa el error en un punto aleatorio, conocido como semilla, obteniendo un vector que indica la pendiente de la función, y, por tanto, la dirección en la que el error aumenta (∇F). Dado que el problema es multidimensional (hay múltiples parámetros que pueden influir sobre la respuesta), es necesario calcular el error en derivadas parciales (ecuación C.4):

$$\nabla F = \begin{pmatrix} \frac{\partial \text{error}}{\partial \Theta_1} \\ \frac{\partial \text{error}}{\partial \Theta_2} \end{pmatrix} \quad (\text{C.4})$$

Teniendo en cuenta que cada parámetro (n) representa una dimensión, para dibujarlo sería necesario un gráfico n -dimensional. Aunque el problema puede tener muchas dimensiones, en la figura C.5 se muestra un gráfico con solo tres dimensiones por sencillez. Todas estas derivadas parciales conforman el ∇F , que indica la dirección en la que la pendiente asciende. Con esta información, el algoritmo se mueve en sentido contrario reduciendo así el error del modelo.

Cuando por fin se alcanza una pendiente cercana a cero, probablemente el sistema haya encontrado un mínimo local. Lo que avanza el algoritmo en cada iteración se conoce como factor de aprendizaje (α), y es crucial para alcanzar una buena

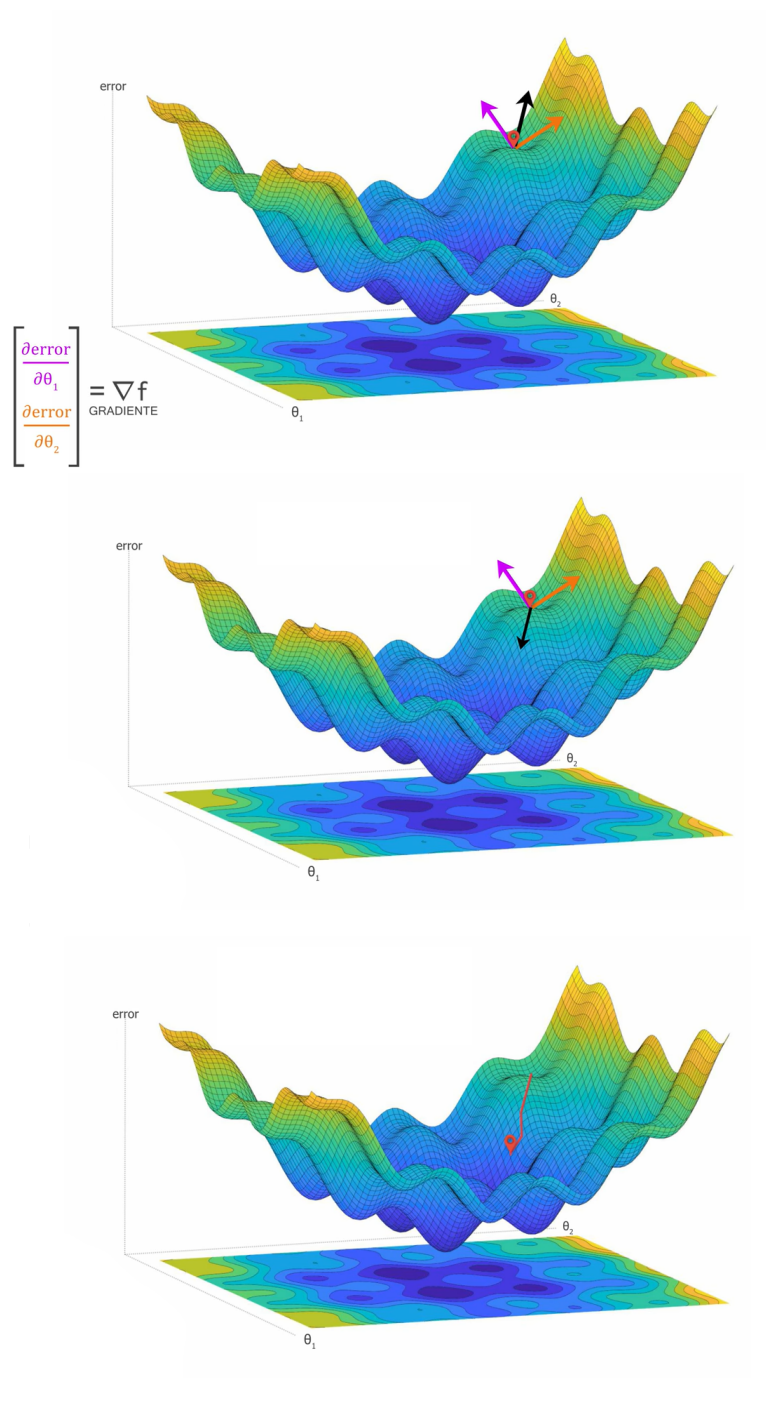


Figura C.5: Gráfica 3D del proceso del algoritmo descenso del gradiente.

solución, pues si es muy grande, puede no llegar nunca a un mínimo local, es decir, la red no convergería nunca (figura C.6). En caso contrario, si α es muy pequeña, podría pasar que la convergencia fuera lenta e ineficiente, o que no se produzca nunca, porque se atasca en un mínimo local más ineficiente (figura C.6).

Cuando se trabaja con regresión lineal simple ($y = W_1X + b$) solo hay dos parámetros que afecten a la predicción del modelo, con lo que para conocer cómo cambia el error de la función respecto a cada uno de los parámetros basta con realizar las derivadas parciales de la función de error respecto a cada parámetro. Sin embargo, cuando se trabaja con redes neuronales, la forma en la que la variación de un parámetro puede afectar al resultado de la red neuronal es más complejo (figura C.7). En esta figura se observa que el parámetro Θ puede afectar al resultado final dependiendo del camino que se estudie, y hay tantas posibilidades como conexiones entre neuronas.

Para facilitar el entendimiento y la reducción del error se utiliza el algoritmo de retropropagación del error (*backpropagation*). El algoritmo de *backpropagation* utiliza pues, el descenso de gradiente para minimizar el error, y propaga el error hacia atrás asignando parte de la culpa de ese error a cada neurona y ajustando los pesos en consecuencia (figura C.8).

El aprendizaje automático puede dividirse en tres grupos diferentes [30]:

- Aprendizaje supervisado: el modelo predictivo es generado basado en datos de entrada y salida. El aprendizaje se considera supervisado ya que se sabe cuál es la salida correcta ante una entrada dada, y la función de la red neuronal es reconocer qué patrones esconden las relaciones entrada – salida, y ser capaz de extrapolar ese aprendizaje en la predicción de nuevas entradas que no ha visto nunca antes. Ese conjunto de entradas – salidas conocidas compone los conjuntos de entrenamiento (un 80 % de los datos) de test (20 % de los datos). Dentro de este grupo se engloban los siguientes métodos: k-vecinos más cercanos, redes neuronales artificiales (ANNs), máquinas de soporte vectorial, clasificador bayesiano, árboles de decisión y regresión logística.
- Aprendizaje no supervisado: estos algoritmos trabajan de una forma similar a los anteriores, con la diferencia de que no se participa activamente en la supervisión del aprendizaje de la máquina. Es decir, a diferencia del supervisado, los datos de entrada no están clasificados ni etiquetados, y no son necesarias estas características para entrenar el modelo. Dentro de este tipo de algoritmos, el agrupamiento o *clustering* en inglés es el más utilizado, ya que divide los datos en grupos que posean características similares. Una aplicación es el reconocimiento de objetos como por ejemplo una silla, o en

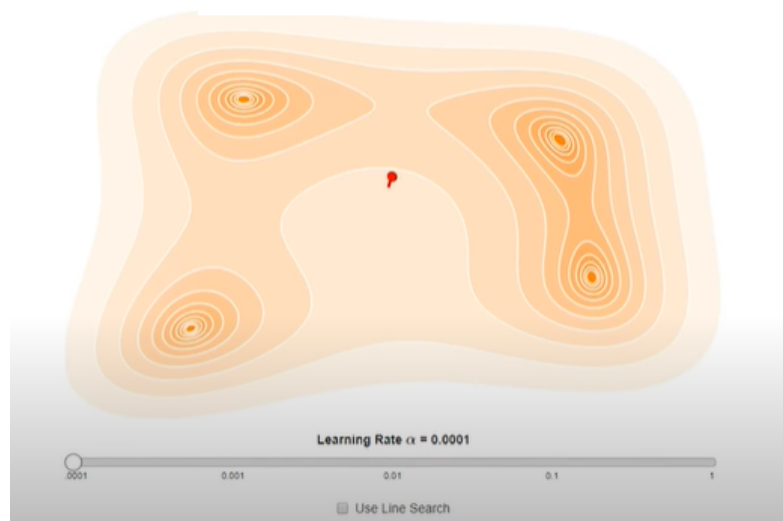
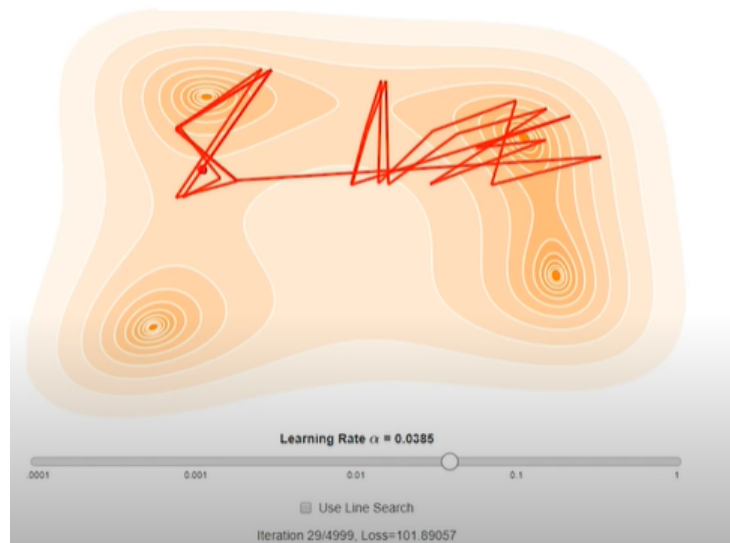


Figura C.6: Gráfico topológico 2D del proceso del algoritmo descenso del gradiente. En la figura de arriba el factor de aprendizaje es muy elevado y nunca se alcanza el mínimo error permitido. Abajo, el factor de aprendizaje es muy pequeño y el coste computacional del algoritmo es inasumible.

el ámbito de la medicina, un tumor. Si las variables de entrada son imágenes de personas, los algoritmos de aprendizaje no supervisado son capaces de

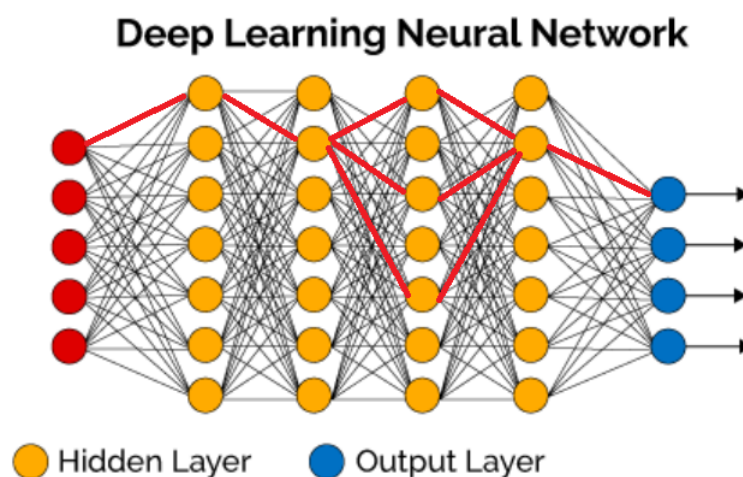


Figura C.7: Diferentes conexiones que pueden estar relacionadas con una salida.

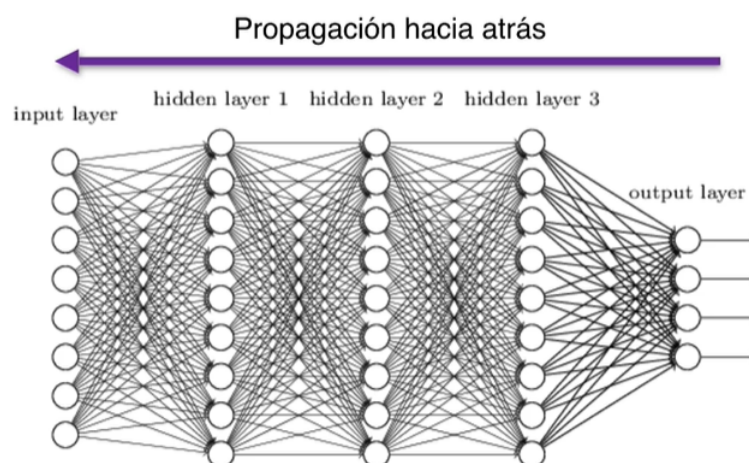


Figura C.8: Retropropagación del error.

agrupar elementos como que las personas generalmente tienen dos piernas, dos brazos, una cabeza y un tronco, siendo capaces de extrapolar esa información para predecir, por ejemplo, dónde hay personas cuando un coche va conduciendo de forma autónoma por una carretera. Entre los principales algoritmos de tipo no supervisado destacan: k-medias, mezcla de gaussianas, agrupamiento jerárquico o mapas auto-organizados.

- Aprendizaje reforzado: Los algoritmos de aprendizaje por refuerzo definen modelos y funciones enfocadas en maximizar una medida de “recompensas”,

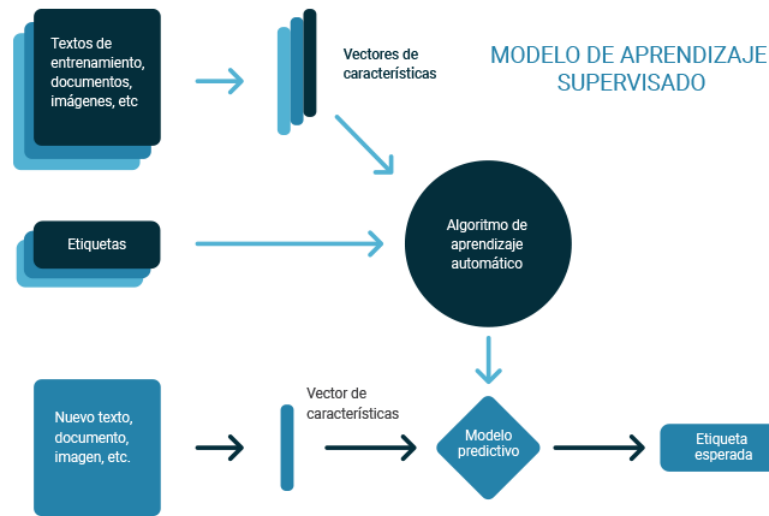


Figura C.9: Diagrama de flujo del aprendizaje supervisado.

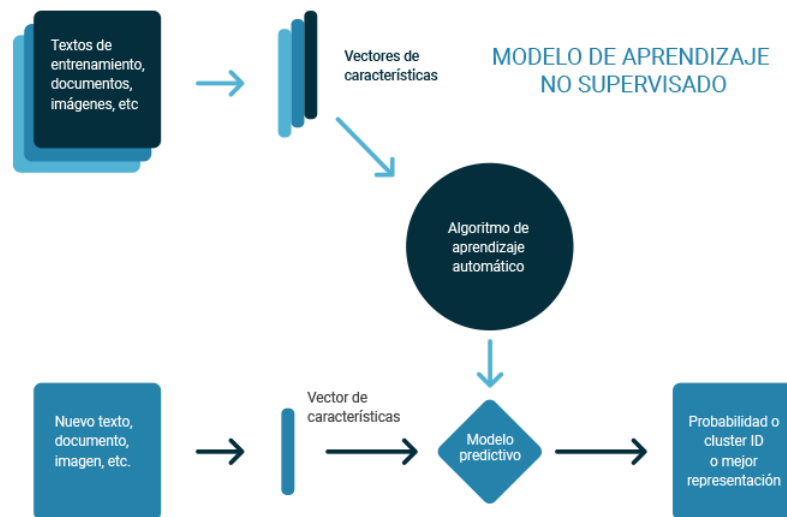


Figura C.10: Diagrama de flujo del aprendizaje no supervisado.

basados en “acciones” y al ambiente en el que el agente inteligente se desempeñará. Este algoritmo es el más apegado a la psicología conductista de los humanos, ya que es un modelo acción-recompensa, que busca que el algoritmo se ajuste a la mejor “recompensa” dada por el ambiente, y sus acciones están sujetas a estas recompensas. Este tipo de métodos pueden usarse para hacer que los robots aprendan a realizar diferentes tareas. Entre los algoritmos más utilizados destacan: programación dinámica, q-learning y SARSA

MODELO DE APRENDIZAJE POR REFUERZO



Figura C.11: Diagrama de flujo del aprendizaje por refuerzo.

C.2. Selección de librerías

El entorno de programación que se ha utilizado ha sido Python 3.8. La ventaja de Python, además de ser simple y rápido, es que tiene una gran comunidad que lo utiliza. Esto implica que se han desarrollado multitud de librerías que facilitan el trabajo a los usuarios, lo que se traduce en un ahorro de tiempo y permite poner el foco en resolver nuevos problemas.

Una librería es un conjunto de rutinas o funciones escritas en un código determinado. El uso de librerías permite ahorrar líneas de código y por tanto crear un entorno más ergonómico para el desarrollo de nuevas herramientas. La selección de librerías se realiza teniendo en cuenta el problema que se va a estudiar, la habilidad del programador, y la necesidad de personalización. A continuación, se presentan las librerías utilizadas divididas en grupos según su tipología:

- Librerías básicas para la ciencia de datos: Pandas, diseñada para trabajar con datos etiquetados; Numpy, se refiere a *Numerical Python*, es esencial para vectorizar operaciones matemáticas mejorando el rendimiento y realizar operaciones matriciales complejas
- Librerías para visualizar datos: Matplotlib, sirve para crear diagramas en 2D

- Librerías de *Machine Learning*: Scikit-learn: interfaz que recoge diferentes algoritmos de ML; Tensor-Flow: permite crear redes neuronales a alto nivel. Los datos se organizan en tensores.

C.3. Escalado de variables

Cuando los valores numéricos de dos parámetros diferentes no son del mismo orden puede haber problemas, ya que los algoritmos de ML utilizan la distancia euclídea (C.5) para comparar parámetros. Sin embargo, cuando el rango de valores difiere entre parámetros, al elevar al cuadrado esos números la diferencia es aún mayor, haciendo que el efecto del parámetro de menor orden pueda llegar a pasar inadvertido. Por ello, el escalado de variables es fundamental. Existen varios tipos de escalados, pero en este trabajo se ha utilizado el escalado máximo - mínimo. Su función es convertir el dato más pequeño del conjunto de datos a cero, el máximo a uno, y el resto quedan interpolados entre ambos valores.

$$DistanciaEuclidea = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (C.5)$$

donde x_i y y_i hacen referencia a las coordenadas x e y del parámetro en el ejes cartesianos.

C.4. Funciones de activación

La función de activación juega un papel fundamental en las redes neuronales, pues es la encargada de aportar la no linealidad a la concatenación de regresiones lineales que se producen de al encadenar neuronas de manera secuencial. Dado que el problema que se trata en este trabajo es un problema de regresión, las funciones de activación más apropiadas son:

- Lineal: la salida es directamente proporcional a la entrada. Es una función derivable.

$$f(x) = ax \quad (C.6)$$

- Sigmoide: satura los valores grandes positivos en 1 y los negativos en cero

$$f(x) = \frac{1}{1 + e^{-x}} \quad (C.7)$$

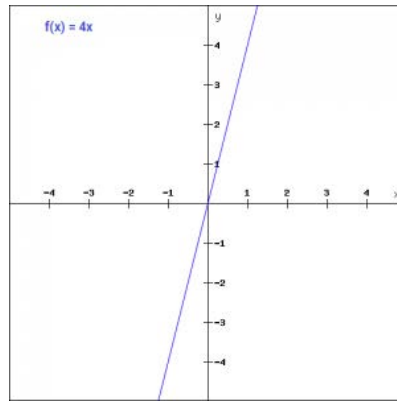


Figura C.12: Función de activación lineal.

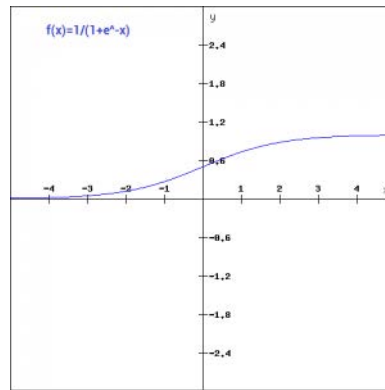


Figura C.13: Función de activación sigmoide.

- Tangente hiperbólica: es similar a la sigmoide pero es simétrica respecto al origen.

$$f(x) = 2 \cdot \text{sigmoid}(2x) - 1 \quad (\text{C.8})$$

- ReLU: Las funciones de Rectificado lineal son transformaciones que activan un nodo sólo si la entrada está por encima de una cierta cantidad. Mientras la entrada es inferior a cero, la salida es cero, pero cuando la entrada supera un cierto umbral, tiene una relación lineal con el dependiente variable

$$f(x) = \max(0, x) \quad (\text{C.9})$$

- Leaky ReLU: es una versión mejorada de la ReLU, que puede tomar valores

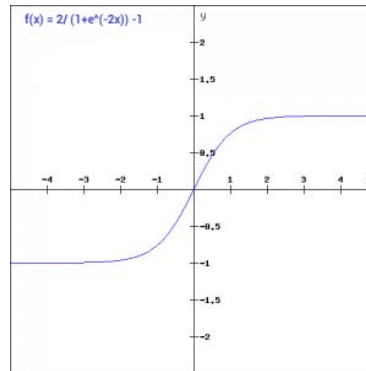


Figura C.14: Función de activación tangente hiperbólica.

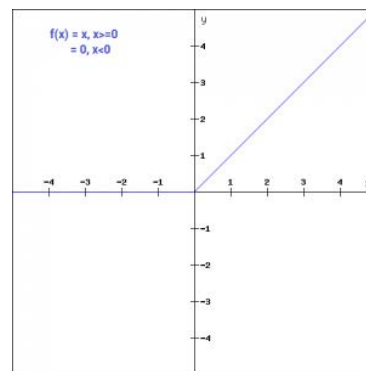


Figura C.15: Función de activación ReLU.

negativos

$$f(x) = \begin{cases} 0,01x & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases} \quad (\text{C.10})$$

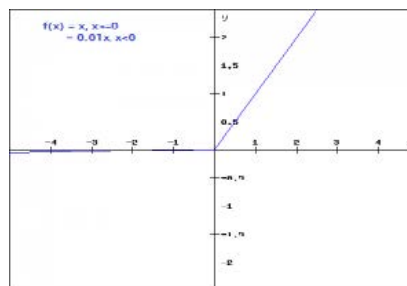


Figura C.16: Función de activación Leaky ReLU.

- Elu: unidad lineal exponencial, es otra variante de la ReLU que introduce una curva exponencial para los valores negativos x; con una asíntota horizontal en -1.

$$f(x) = \begin{cases} a(e^x - 1) & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases} \quad (\text{C.11})$$

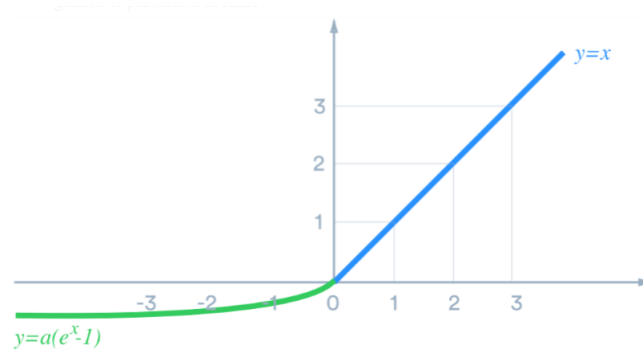


Figura C.17: Función de activación Elu.

- Selu: es como la Elu pero escalada. Es útil porque hará que la salida de cada capa conserve la media 0 y desviación estándar 1 durante el entrenamiento.

$$f(x) = \begin{cases} scale \cdot a(e^x - 1) & \text{si } x < 0 \\ scale \cdot x & \text{si } x \geq 0 \end{cases} \quad (\text{C.12})$$

- Softmax: sirve para extraer la probabilidad de pertenecer a distintas clases

$$\theta(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (\text{C.13})$$

- Softplus: Las funciones de activación Softplus se consideran la versión suavizada de las transformaciones por rectificación lineal. Si bien la forma es similar, vemos que la función que las define es derivable en todo su dominio

$$\phi(x) = \ln(1 + e^x) \quad (\text{C.14})$$

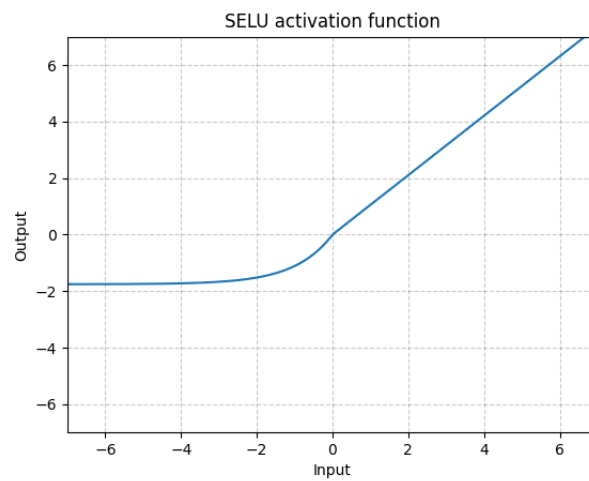


Figura C.18: Función de activación Selu.

Anexo D

Código completo de la Red neuronal

```
# Red Neuronal Artificial
# Ricardo Caballero Masa
# 28/05/2021

# importar las librerias
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import BatchNormalization

# Importar el data set
mis_datos = pd.read_excel('DataSet_Inc14.xlsx')
mis_datos = mis_datos.to_dict("list")
x_vals = np.transpose(np.array([mis_datos['E_Core_cuad'],
                                mis_datos['E_Fib_Cap_cuad'], mis_datos['E_Theta_cuad'],
                                mis_datos['Estenosis(%)']]))
y1_vals = np.transpose(np.array([mis_datos['Ecore']]))
y2_vals = np.transpose(np.array([mis_datos['Eplaca']]))
y_vals = np.transpose((y1_vals[:,0],y2_vals[:,0]))

#Me aseguro de que el data set es el que quiero
print(x_vals.shape[1])
print(y_vals.shape)
```

```

# Dividir el data set en conjunto de entrenamiento y conjunto de
testing

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x_vals, y_vals
    , test_size = 0.2, random_state = 0)

# Escalado de variables
from sklearn.preprocessing import MinMaxScaler
sc_X = MinMaxScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)

# Crear la RNA
my_ANN = Sequential()
my_ANN.add(Dense(units = 250, kernel_initializer = "lecun_normal"
    , activation = "softplus", input_dim = x_vals.shape[1]))

#Incorporamos las capas ocultas
my_ANN.add(Dropout(0.05))
my_ANN.add(Dense(units = 150, kernel_initializer = "lecun_normal",
    activation = "softplus"))
my_ANN.add(BatchNormalization(axis=1))
my_ANN.add(Dense(units = 100, kernel_initializer = "lecun_normal",
    activation = "softplus"))

my_ANN.add(Dense(units = 50, kernel_initializer = "lecun_normal",
    activation = "softplus"))

my_ANN.add(Dense(units = 40, kernel_initializer = "lecun_normal",
    activation = "softplus"))

my_ANN.add(Dense(units = 30, kernel_initializer = "lecun_normal",
    activation = "softplus"))

my_ANN.add(Dense(units = 10, kernel_initializer = "lecun_normal",
    activation = "softplus"))

my_ANN.add(Dense(units = 2, kernel_initializer = "lecun_normal",
    activation = "softplus"))

```

```

# Entrenar la RNA
my_ANN.compile(loss='mean_squared_error',
               optimizer='adam',
               metrics=['mae'])

my_ANN.fit(X_train, y_train, batch_size = 8, epochs=300)

# Predecir con los datos reservados para test
# y comparar los resultados de la prediccion con los reales
np.set_printoptions(suppress = True) #quitamos la notacion
cientifica
yhat = my_ANN.predict(X_test)
print(yhat)
print(y_test)

# Exportar datos a un fichero excel
from pandas import ExcelWriter
df = pd.DataFrame({'Ecore_predict': yhat[:,0],
                  'Eplaca_predict': yhat[:,1],
                  'Ecore_test': y_test[:,0],
                  'Eplaca_test': y_test[:,1]})
df = df[['Ecore_predict', 'Eplaca_predict', 'Ecore_test', 'Eplaca_test']]
writer = ExcelWriter('ejemplo.xlsx')
df.to_excel(writer, 'Hoja de datos', index=False)
writer.save()

#Prediciendo datos correspondientes a geometria real
# Importar el data set de VALIDACION
mis_datos3 = pd.read_excel('DataSetGeomReal1.xlsx')
mis_datos3 = mis_datos3.to_dict("list")

x_vals3 = np.transpose(np.array([mis_datos3['E_Core_cuad'],
                                mis_datos3['E_Fib_Cap_cuad'], mis_datos3['E_Theta_cuad'],
                                mis_datos3['Estenosis(%)']]))
y1_vals3 = np.transpose(np.array([mis_datos3['Ecore']]))
y2_vals3 = np.transpose(np.array([mis_datos3['Eplaca']]))
y_vals3 = np.transpose((y1_vals3[:,0], y2_vals3[:,0]))

x_vals_total = np.append(x_vals, x_vals3, axis=0)
y_vals_total = np.append(y_vals, y_vals3, axis=0)

# Pongo los nuevos datos en la anterior base de datos para que al
  escalarlos

```

```

# y que tengan sentido. Si los escalamos sin ser incluidos en esta
# base de datos los
# escalara a cero. Ademas se pone axis=0 para que lo incorpore al
# array anterior
# en la posicion correcta (por filas)

print(x_vals3)
print(y_vals3)

# Control de que los datos son correctos
print(x_vals_total)
print(y_vals_total)

# Escalado de variables
from sklearn.preprocessing import MinMaxScaler
sc_X = MinMaxScaler()
x_vals_total = sc_X.fit_transform(x_vals_total)
print(x_vals_total)

# Control de que los datos son correctos
print(x_vals_total[-1:])

# Predecir con los datos reservados para validacion
# y comparar los resultados de la prediccion con los reales
np.set_printoptions(suppress = True) #quitamos la notacion
cientifica
yhat3 = my_ANN.predict(x_vals_total[-1:])
print(yhat3)
print(y_vals3)

# Guardar modelo entrenado
import joblib
joblib.dump(my_ANN, 'modeloentrenado_V4.pkl')

### comando para cargar el modelo:
import joblib
my_ANN = joblib.load('modeloentrenado_V4.pkl')

```